

COMPUTATIONAL GEOMETRY USING CONTROL GEOMETRY HAVING AT LEAST TWO DIMENSIONS

CROSS-REFERENCE TO RELATED APPLICATIONS

5 The present application is a divisional of U.S. Patent Application Serial No. 09/360,029 filed July 23, 1999, which claims priority from U.S. Provisional Application Serial No. 60/093,892, filed July 23, 1998, and from U.S. Provisional Application Serial No. 60/116,199, filed January 15, 1999, all of which are incorporated herein by reference in their entirety.

FIELD OF THE INVENTION

The present invention relates to a system and method for performing computer aided design, and, in particular, to efficient computational techniques for blending between representations of geometric objects.

10 BACKGROUND

A designer using a computer aided design (CAD) computational system will typically approach the design of a free form geometric object (such as a surface) by first specifying prominent and/or necessary subportions of the geometric object through which the object is constrained to pass. Subsequently, a process is activated for generating the geometric object that conforms to the constraining subportions provided. In particular, such subportions may be points, curves, surfaces and/or higher dimensional geometric objects. For example, a designer that designs a surface may construct and position a plurality of curves through which the intended surface must pass (each such curve also being denoted herein as a "feature line" or "feature curve"). Thus, the intended surface is, in general, expected to have geometric characteristics (such as differentiability and curvature) that, substantially, only change to the extent necessary in order to satisfy the constraints placed upon the surface by the plurality of curves. That is, the designer expects the generated surface to be what is typically referred to as "fair" by those skilled in the art. 15 Thus, the designer typically constructs such feature curves and positions them where the intended surface is likely to change its geometric shape in a way that cannot be easily interpolated from other subportions of the surface already designed. 20

As a more specific example, when designing containers such as bottles, an intended exterior surface of a bottle may be initially specified by subportions such as: (a) feature curves positioned in high curvature portions of the bottle surface, and

(b) surface subareas having particular geometric characteristics such as having a shape or contour upon which a bottle label can be smoothly applied. Thus, the intention of a bottle surface designer is to construct a bottle design that satisfies his/her input constraints and that is also fair. Moreover, the designer may desire to generate holes for handles, as well as, e.g., ergonomic bottle grips by deforming various portions of the bottle surface and still have the bottle surface fair.

5 There has heretofore, however, been no CAD system wherein a designer (or more generally, user) of geometric objects can easily and efficiently express his/her design intent by inputting constraints and having the resulting geometric object be fair. That is, the designer/user may encounter lengthy delays due to substantial computational overhead and/or the designer/user may be confronted with non-intuitive geometric object definition and deformation techniques that require substantial experience to effectively use. For example, many prior art CAD systems provide techniques for allowing surfaces to be designed and/or deformed
10 by defining and/or manipulating designated points denoted as “control points.” However, such techniques can be computationally expensive, non-intuitive, and incapable of easily deforming more than a local area of the surface associated with such a control point. Additionally, some prior art CAD systems provide techniques for defining and/or deforming surfaces via certain individually designated control vectors. That is, the direction of these vectors may be used to define the shape or contour of an associated surface. However, a designer’s intent may not easily correspond to a surface design technique using such control vectors since
15 each of the control vectors typically corresponds to only a single point of the surface isolated from other surface points having corresponding control vectors. Thus, such techniques are, at most, only able to deform an area of the surface local to such points having corresponding control vectors.

 Additionally, such prior art CAD systems may also have difficulties in precisely performing blending and trimming operations. For example, two geometric objects intended to abut one another along a common boundary may not be within a
20 sufficient tolerance to one another at the boundary. That is, there may be sufficiently large gaps between the geometric objects that the boundary may not be considered “water tight,” which may be problematic in certain machining operations and other operations like Boolean operations on solids.

Accordingly, it would be very desirable to have a CAD system that includes one or more geometric design techniques for allowing CAD designers/users to more easily, efficiently and precisely design geometric objects. Further, it would be desirable to have such a system and/or computational techniques for graphically displaying geometric objects, wherein there is greater user control over the defining and/or deforming of computational geometric objects, and in particular, more intuitive global control over the shape or contour of computationally designed geometric objects.

Definitions

This section provides some of the fundamental definitions that are used in describing the present invention. These definitions are also illustrated in Figs. 15 and 16.

A “*parametric geometric object*” S is a geometric object that is the image of a function f , wherein the domain of f is in a geometric space embedded within a coordinate system (denoted the “*parameter space*”) and the range of f is in another geometric space (denoted the “*object space*”). Typically, the inverse or pre-image, f^{-1} , of a geometric object such as S will be a geometrically simpler object than its image in object space. For example, the pre-image of a curve 170 in object space may be a simple line segment 172, L , in parameter space. Thus if S denotes the curve in object space, then notationally f and S are sometimes identified such that for $u \in L$, a corresponding point in the curve S is denoted $S(u)$. Similarly, the pre-image of an undulating surface 204 (Fig. 16) in object space may be a simple bounded plane 180 in parameter space. Thus, if S denotes the undulating surface 204, then for $(u,v) \in f^{-1}(S)$, $S(u,v)$ denotes a corresponding point on the undulating surface 204.

A “*profile*” 200 (Fig. 16) is a geometric object, such as a curve in object space, through which an associated object space geometrically modeled object (e.g. surface 204) must pass. That is, such profiles 200 are used to generate the geometrically modeled object. Thus, profiles provide a common and natural way for artists and designers to geometrically design objects, in that such a designer may think in terms of the feature or profile curves when defining the characteristic shape of a geometric object (surface) being designed. For example, profile curves on a surface may substantially define the geometry of a resulting derived geometric object; e.g., its continuity, curvature, shape, boundaries, kinks, etc. Note, that for many design applications,

profiles are typically continuous and differentiable. However, such constraints are not necessary. For example, a profile may, in addition to supplying a general shape or trend of the geometric object passing therethrough, also provide a texture to the surface of the geometric object. Thus, if a profile is a fractal or fractal-like, the fractal contours may be in some measure imparted to the surface of the derived geometric object adjacent the profile. Further note that it is within the scope of the present invention to utilize profiles that are of higher dimension (≥ 2). Thus, a profile may also be a surface or a solid. Accordingly, if a profile is a surface, then a solid having locally (i.e., adjacent to the profile) at least some of the geometric characteristics of the profile may be derived.

Moreover, profiles (and/or segments thereof) may have various computational representations such as linear (e.g., hyperplanes), elliptic, NURBS, or Bezier. Note, however, that regardless of the computational representation, a method (such as interpolation) for deforming or reshaping each profile is preferable. More particularly, it may be preferable that such a method results in the profile satisfying certain geometric constraints such as passing through (or substantially so) one or more predetermined points, being continuous, being differentiable, having a minimal curvature, etc. Further, note that such a deformation method may also include the ability to decompose a profile into subprofiles, wherein the common boundary (e.g., a point) between the subprofiles may be “slidable” along the extent of the original profile.

A “**marker**”²⁰⁸ (Fig. 16) is a point on a profile that can be moved to change the shape of the profile ²⁰⁰ in a region about the marker. A marker also designates a position on a profile where the shape of a geometric object having the profile thereon can be deformed.

A “**profile handle**”²¹² (Fig. 16) is a geometric object tangent to the profile ²⁰⁰. Such a profile handle may control the shape of the profile locally by modifying the slope (derivative) of the profile at the marker ²⁰⁸. Alternatively, for non-differentiable profiles, a profile handle may be used to control the general shape of the profile by indicating a trend direction and magnitude of the corresponding profile. For example, if the profile is a fractal or other nondifferentiable geometric object, then a profile handle may, for example, provide a range within the object space to which the profile must be confined; i.e., the range

may be of a tubular configuration wherein the profile is confined to the interior of the tubular configuration, Note that the profile handle 212 affects the fullness of the profile 200 (e.g., the degree of convexity deviating from a straight line between markers on the profile) by changing the length of the profile handle.

5 An *"isocline boundary"* 220 is the boundary curve opposite the profile 200 on the isocline ribbon 216. In one embodiment, at each point on the profile 200 there is a paired corresponding point on the isocline boundary 200, wherein each such pair of points defines a vector 224 (denoted a **"picket"**) that is typically transverse to a tangent vector at the point on the profile. More particularly, for a parameterized profile, the isocline boundary 220 can be viewed as a collection of pickets at all possible parameter values for the profile 200.

10 An *"isocline ribbon"* (or simply isocline) is a geometric object, such as a surface 216, which defines the slope of the geometric object (e.g., surface) 204 (more generally a geometric object 204) at the profile 200. Equivalently, the isocline ribbon may be considered as the representation of a geometric object delimited by the profile 200, the isocline handles 218a and 218b (discussed hereinbelow), and the isocline boundary 220. In other words, the geometric object 204 must **"heel"** to the isocline ribbon 216 along the profile 200. Said another way, in one embodiment, the geometric object 204 must be continuous at the isocline 216 and also be continuously differentiable across the profile 200. In an alternative embodiment, the geometric
15 object 204 may be constrained by the isocline 216 so that the object 204 lies within a particular geometric range in a similar manner as discussed above in the description of the term "profile." Note that there may be two isocline ribbons 216 associated with each profile 200. In particular, for a profile that is a boundary for two abutting surfaces (e.g. two abutting surfaces 204), there can be an isocline ribbon along the profile for each of the two surfaces. Thus we may speak of a right and a left hand isocline ribbon.

20 An *"isocline handle"* 228 is a geometric object (e.g., a vector) for controlling the shape of the isocline ribbon 216 at the marker 208, wherein the profile handle and isocline handle at the marker may define a plane tangent to the surface 204. Hence the isocline handle is used to determine the shape of the surface 204 (or other underlying geometric object) about the

marker. In particular, an isocline handle 228 is a user manipulatable picket 224. If all the profile handles 212 and isocline handles 228 (e.g., for two or more abutting surfaces) are coplanar at a marker 208, then the surface 204 will be smooth at the marker (assuming the surface is continuously differentiable), otherwise the surface may have a crease or dart. Note that by pulling one of the handles (either isocline or profile) out of the plane of the other handles at a marker, one may intentionally generate a crease in the surface 204 along the profile 200.

The part of the profile 200 between two markers 208 is denoted a *“profile segment”*²³². Similarly, the part of the isocline ribbon 216 between two isocline handles 228 is denoted a *ribbon segment*²⁴⁰.

A *“boundary segment”*²⁴⁴ denotes the part of the boundary 220 between two isocline handles 228.

The vector 246 that is the derivative tangent to the isocline boundary 220 at an isocline handle 228 is denoted a **“ribbon tangent.”** Note that modifications of ribbon tangents can also be used by the present invention to control and/or modify the shape of an underlying geometric object such as surface 204.

Isocline handles 228 may be generalized to also specify curvature of the surface 204. That is, instead of straight vectors as isocline handles, the handles may be curved and denoted as *“isocline ribs”*²⁴⁸. Thus, such ribs may facilitate preserving curvature continuity between surfaces having associated isocline ribbons along a common profile boundary, wherein the isocline ribbons are composed of isocline ribs. Accordingly, the curvature of such surfaces will match the curvature of their corresponding isocline ribs, in much the same way as they match in tangency.

A **“developable surface”** is a surface that can be conceptually rolled out flat without tearing or kinking. It is a special case of a **“ruled surface,”** this latter surface being defined by being able to lay a ruler (i.e., straight edge) at any point on the surface and find an orientation so that the ruler touches the surface along the entirety of the ruler. For a developable surface, the surface perpendiculars are all equal in direction along the ruling.

“Label surfaces” denote special 2-dimensional (developable or nearly developable) surfaces wherein a label may be applied on, e.g., a container. Label surfaces allow application of a decal without tearing or creasing. These surfaces are highly constrained and are not typically deformed by the geometric modification of an isocline ribbon 216.

5 A **“trim profile”** is a geometric object (curve) that is a profile for trimming another geometric object (e.g., a surface). The trim profile may have a single corresponding isocline ribbon 216 since if the surface to be trimmed is a label surface, it will not be modified and, accordingly, no isocline ribbon can be used to change its shape.

A trim profile (curve) can be used to delimit any surface, not just a label surface. A surface, S , that is blended along a trim profile with one or more other surfaces is called an **“overbuilt surface”** when the surface S overhangs the trim profile. For example, in Fig. 12, surface 130 is an overbuilt surface, wherein the portion of the surface outside of the area 134 is typically not shown to the designer once it has been trimmed away.

10 A **convex combination** of arguments F_i is a summation $\sum_i c_i F_i$ where the c_i are scalar coefficients and scalar multiplication is well-defined for the F_i (e.g., F_i being vectors, functions, or differential operators), and where $c_i \geq 0$ and $\sum_i c_i = 1$. If the F_i are points in space, for instance, then the set of all possible such combinations yields the convex hull of the points F_i , as one skilled in the art will understand.

15 A **“forward evaluation”** is a geometric object evaluation technique, wherein in order to generate a set of sample values from a function, $f(x)$, argument values for x are incremented and f is subsequently evaluated. This type of evaluation is usually fast and efficient, but does not give function values at chosen positions between the increments.

20 An **“implicit function”** is one written in the form $f(x)=0$. $X \in \mathbb{R}^n$ When a parametric curve or surface is converted to an implicit form, the conversion is called **“implicitization.”** Hence $f(t)=(\sin(t), \cos(t))$ in parametric form may be implicitized by $f(x,y) = x^2 + y^2 - 1 = 0$. Both forms describe a circle.

Dividing a vector by its length **“normalizes”** it. The normalized vector then has unit length. A vector function may be divided by its gradient, which will approximate unit length, as one skilled in the art will understand.

Given a function defined by a $\sum_i p_i(t) F_i(t)$ where $p_i(t)$ are weighting functions, if $\sum_i p_i(t) = 1$ for all values of t , then the p_i are said to form a “**partition of unity**.”

“**GI**” continuity denotes herein a geometric continuity condition wherein direction vectors along a continuous parametric path on a parametrically defined geometric object are continuous, e.g., tangent vector magnitudes are not considered.

5

SUMMARY OF THE INVENTION

The present invention is a computational geometric design system that is capable of sufficiently efficient computations so as to allow real-time deformations to objects such as surfaces while a user is supplying the object modifying input. Thus, the present invention is a paradigm shift away from typical CAD systems since, in a typical CAD system the user must supply input for changing or modifying a geometric object and subsequently explicitly request processing of the input to commence. Thus, in
10 such prior art CAD systems, the user receives feedback about his/her design at discrete user requested times. Instead, with the present invention, updates may be processed in real-time immediately upon input receipt without the user explicitly indicating that update processing is to be performed.

Given the enhanced computational efficiency of the present invention, a user of the present invention can more efficiently perform iterative approximations to a geometric object being designed. The user may speedily design without the need
15 to precisely calculate design geometric characteristics for portions of the object where such precision may not be necessary. That is, the user can be less concerned about getting it "right the first time" since the ease of modification and speed of computing modifications allows the user to more rapidly approximate and/or prototype a geometric object. Thus, the present invention can have substantial efficiency benefits in that for many geometrically designed objects (including machined parts), substantial portions of such objects may be satisfactorily designed with a wide range of geometric characteristics.

20 The CAD system of the present invention enables novel design techniques by providing a novel computational technique for blending between two parametric geometric objects such as surfaces. In one embodiment of the present invention, this novel blending technique blends between two parametric surfaces $S_1(u,v)$ and $S_2(u,v)$, wherein each surface has, e.g., the unit square

as its parameter space. Assuming each surface S_1 and S_2 has a respective blending function $B_1(u,v)$ and $B_2(u,v)$ such that each of the blending functions has, for example, $(0,1)$ as its range for u and v (as well as satisfying other properties given hereinbelow), a new surface, S , may be defined by the following formula:

$$S(u, v) = S_1(u, v) \cdot B_1(u, v) + S_2(u, v) \cdot B_2(u, v) \quad (1)$$

Note that the blending functions B_1 and B_2 are typically chosen so that the resulting blended surface S is the same as S_1 on a boundary with S_1 , and the same as S_2 on a boundary with S_2 . This is achieved by devising B_1 and B_2 so that $B_1=1$ and $B_2=0$ on the boundary with S_1 while having $B_2=1$ and $B_1=0$ on the boundary with S_2 .

In a more general embodiment, the present invention may be used for blending between a plurality of geometric objects, S_i , $i=1, \dots, N$, wherein each of the geometric objects is parameterized by a corresponding function f_{S_i} whose domain includes a parameter space PS common to all f_{S_i} . Thus, for the resulting blended surface S , substantially every one of its points, $S(q)$, for q in PS is determined using a weighted sum of points obtained from the points $f_{S_i}(q)$. Moreover, since it is desirable to blend S to a boundary portion P_i of each S_i , when interpreting S as a function from PS to the common geometric space GS having the geometric objects S_i , then $S(f_{S_i}^{-1}(P_i)) \subseteq P_i$. Additionally, S may be also continuous at each $f_{S_i}^{-1}(P_i)$.

Note that Formula (1) above is representative of various formulas for generating a blended surface (more generally, geometric object) S , other embodiments of such formulas are provided in the Detailed Description hereinbelow. Further note that such formulas may be generalized wherein the parameter space coordinates (u,v) of Formula (1) above can be replaced by representations of other parameter space coordinates such as triples (u,v,w) or merely a single coordinate u . Moreover, the blending functions B_1 and B_2 may also be defined for such other parameter spaces. Additionally, it is worthwhile to note that such blending functions B_1 and B_2 may be considered as weights of a weighted sum of points selected from the surfaces (more generally, geometric objects) S_1 and S_2 . Further, this weighted sum notion may also be extended in various ways. For example, referring

to the more general embodiment wherein a plurality of geometric objects S_i , $i=1, \dots, N$ are provided, a corresponding weight/blending function B_i may be provided for each value of i so that the following variation of Formula (1) is obtained:

$$S(q) = \sum_{i=1}^N B_i(q) S_i(q) \text{ for points } q \text{ in a common parameter space for the } S_i.$$

In another aspect of the present invention, it is within the invention scope to also generate blended geometric objects S , wherein at least some of the S_i geometric objects used to generate S are of a higher dimension than 2. For example, if S_1 and S_2 are parameterized solids, then S may be generated as a solid blended from S_1 and S_2 using another variation of Formula (1), as one skilled in the art will understand. Thus, S may extend between S_1 and S_2 so that a surface P_1 of S_1 and a surface P_2 of S_2 are also boundaries of S , and S is represented as a weighted sum of points of S_1 and S_2 similar to Formula (1).

In yet another aspect of the present invention, one or more of the parametric geometric objects S_i of Formula (1) (or variations thereof such as Formula (2), (4), (5), (5.02), (5.03), etc. provided in the Detailed Description hereinbelow) may have one of: a Bezier, NURBS, or some other multivariant parametric computational representation instead of, e.g., isocline ribbons as illustratively used in the description herein. Moreover, as one skilled in the art will also understand, it is within the scope of the present invention that the underlying geometric objects that define the S_i 's (e.g., for the S_i being isocline ribbons, such underlying geometry being markers, profiles, isocline handles and profile handles) may be different for a different computational representation. For example, in a Bezier or NURBS representation of an S_i "control points" and/or geometric entities derived therefrom, may be used to change a shape of the S_i and therefore change the shape of the resulting geometric object S derived therefrom.

In another aspect of the present invention, when a blended surface S is generated from one or more isocline ribbons S_1, \dots, S_N , $N \geq 2$, the surface S can be deformed by changing geometric characteristics of the isocline ribbons S_i . In particular, by changing the shape of one of the isocline ribbon boundaries for some S_i , the points $S_i(u,v)$ change and accordingly, the blended surface S changes since it is a weighted sum of such points. In particular, rates of change of geometric characteristics of S (such as curvature, tangent vectors, and/or tangent planes) may be determined by the shape of the isocline ribbons S_i . More

particularly, assuming a substantially linear parameterization along each isocline picket, the greater the relative magnitude of such pickets for a particular isocline ribbon, the more the shape of S will be skewed in the direction(s) of such pickets. Moreover, as the direction of such pickets changes, the curvature of S changes. That is, since the weighted sums, such as Formula (I), cause S to always heel to the surfaces S_i , the shape of S will change so that S heels to the isocline ribbon(s) S_i having pickets whose directions have changed. Thus, the shape of the blended surface S may be changed by any user interaction technique that: (a) changes one or more geometric characteristics of one or more of the S_i , wherein such changes may include: changing a shape of S_i (wherein shape denotes a plurality of geometric characteristics such as continuity, differentiability, curvature, and higher order continuity), (b) changes a parameterization of an S_i .

Also note that such user interaction techniques for deforming a blended surface may also be used with higher dimensional geometric objects. For example, if the S_i geometric objects are solids rather than surfaces, then a resulting blended solid, S may be deformed by changing a shape of one or more of the solids S_i used in determining S .

It is also within the scope of the present invention that the geometric objects S_i used to generate a blended geometric object S may be such that the S_i 's can be modified indirectly via other geometric objects from which the S_i 's may be themselves generated. For example, if S is a surface blended from isocline ribbons S_1 and S_2 (having corresponding profiles P_1 and P_2 , respectively), and the ribbon S_1 is interpolated from the profile handle, the isocline handle, and the ribbon tangent at the end points of P_1 , then the present invention provides user interaction techniques for modifying such handles and/or ribbon tangents for thereby modifying the blended surface S . Moreover, in one user interface technique, only the handles may be displayed, wherein such handles are displayed as connected to the blended surface S . Thus, by changing such handles, the blended surface changes. Note that such user interaction techniques may be responsive in real time to user changes to such handles and/or ribbon tangents. Thus, a user's design intent may be immediately displayed while the user is inputting such changes. Accordingly, using the present invention, user interactions in the design process may become closer to the techniques in used in constructing actual geometric models rather than prior art CAD user interaction techniques.

It is another aspect of the present invention that various geometric constraint criteria are capable of being applied to geometric objects generated according to the present invention. In particular, features and/or subgeometry of a geometric object O_0 are capable of being constrained to lie within another geometric object, O_1 , so that as O_1 is deformed, the features and/or subgeometry of O_0 deform correspondingly, and thereby cause O_0 to deform accordingly. For example, the present invention allows an object space point p to be defined (i.e., parameterized) so that it must remain in/on a given geometric object O_1 , where O_1 may be a curve, surface, volume or solid. Thus, as O_1 is deformed, O_0 also deforms. Moreover, instead of a point p , other geometric subobjects may also be similarly constrained, such as curves, surfaces or solids. Additionally, features of a geometric object O_0 such as control points, handles (of various types, e.g., profile and isocline), normals, twist vectors, etc. may also be similarly constrained by the present invention so that as O_1 is deformed, O_0 is caused to also deform. For instance, using the geometric object interpolation techniques provided by the present invention, e.g., Formula (1) and variations thereof, the geometric object O_0 can be efficiently regenerated (e.g., reinterpolated) substantially in real-time when constrained features and/or subgeometries of O_0 are correspondingly deformed with a deformation of O_1 . More particularly, this aspect of the present invention provides for the combining of various geometric objects hierarchically so that geometric deformation control of a parent object causes corresponding geometric changes in dependent child geometric objects. For example, when a surface patch represents fine scale detail of a larger surface, it may be advantageous to attach the fine detail surface patch to the larger surface to thereby give a user automatic control over the shape of the fine detail surface patch by controlling the shape of the larger surface. Moreover, similar hierarchical control can be provided with other geometric objects of types such as curves, points and three-dimensional deformation spaces.

Note that such hierarchical control may be also used with a persistent deformation space wherein it is desirable for a geometric object in this space to be repeatedly deformed and restored to its original non-deformed state. Note that this is difficult to do in real-time by repeatedly applying a one-time deformation. Accordingly, by utilizing such hierarchical control of the present invention, a geometric object embedded in such a three-dimensional deformation space and/or the control

structures of the geometric object embedded therein provides for the deformation of the geometric object when the three-dimensional deformation space is deformed. Further, if one or more such deformation spaces are, in turn, made dependent upon a simpler geometry such as a surface or curve, then substantial control over the shape of the geometric object, however complex, can be provided by manipulating the shape of the simpler geometry.

5 Other features and benefits of the present invention are provided in the Detailed Description and the drawings provided herewith.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a surface 62 generated according to the present invention, wherein the surface interpolates between the surfaces 30 and 34, and also passes through the curves 54, 58 and 60 at predetermined directions according to the isocline
10 ribbons 61 and 63;

Fig. 2 shows a further modification of the surfaces of Fig. 1, wherein the surface 30 has a circular disk 66 blended thereto according to a method of the present invention;

Fig. 3 shows a blended surface 62a generated according to the present invention between the surfaces 30 and 34, wherein the surface 62a passes through the curves 54 and 58 and wherein the blending is performed according to a novel surface
15 generation formula provided herein (Formula (1));

Fig. 4 illustrates a correspondence between geometric entities in parameter space and geometric entities in object space, wherein lines 78a and 78b of parameter space have object space images of curve 54 and 58, respectively, and additionally, parameter space line 86 has as an object space image curve 80;

Fig. 5 provides a graphical representation of two blending functions, B_1 and B_2 , utilized in some embodiments of the
20 present invention;

Figs. 6A-6D show graphs of additional blending functions that may be used with the present invention;

Fig. 7 provides a further illustration of the correspondences between geometric entities in parameter space and object space;

Fig. 8 shows an elliptic region 100 that is blended into a cylinder 108 according to the present invention, wherein the closed curve 110 delimits the elliptic region from the deformed portion of the cylinder 108 that blends to the closed curve;

5 Fig. 9 shows a simple boss 112 created on a cylinder 116 according to the method of the present invention;

Fig. 10 shows a composite curve 120 (as defined hereinbelow) that includes two crossing subcurves 124 and 128;

Fig. 11 shows a surface 130 from which a label surface 134 is trimmed;

Fig. 12 illustrates one computational technique for determining a distance-like measurement from a point p that is interior to the polygon having vertices v_1, v_2, v_3, v_4 , and v_5 , wherein the distance-like measurement to each side of this polygon is determined using a corresponding apex 150 provided by a stellating process;

Fig. 13 shows two boundary curves 156a and 156b in parameter space (i.e., the unit square), wherein area patches 168 therebetween are capable of being themselves parameterized by coordinates (s,t) where s varies linearly with the distance between two corresponding points on a pair of opposing subcurves 160a and 160b, and t determines a corresponding point on each of the subcurves of the pair 160a and 160b;

15 Fig. 14 illustrates a region that has sides and ribbons defined by three surfaces S_1, S_2 and S_3 , wherein the present invention is able to provide a surface patch for the region 300 using Formula (5) provided hereinbelow;

Figs. 15 and 16 illustrate both general computational geometry concepts, as well as novel concepts that are fundamental to the present invention. Note that these figures are used to illustrate the terms defined in the Definitions Section hereinabove;

Fig. 17 shows a block diagram of the typical flow of design construction operations performed by a user of the present invention in designing a geometric object;

20 Fig. 18 shows three profile curves 404, 408 and 412 meeting at a profile marker 420, wherein the surfaces 416 and 418 join smoothly at the marker 420 due to the isocline handles (for the marker 420) being in a common plane 460;

Fig. 19 shows profile curves x and y that define a surface 480 which forms a fillet between surfaces 484 and 486. Typically, profiles x and y are defined using distances 488 and 490 from the intersection curve 482 of surfaces 484 and 486;

Fig. 20 illustrates one embodiment for computing a blended surface from isocline ribbons 508 and 516 according to the present invention;

5 Figs. 21A - 21C illustrate a procedure for creating a hole 600 according to the present invention;

Fig. 22 shows a blended surface 710 according to the present invention, wherein the blended surface extends between a degenerate profile (point) 714 and the circular end 718 of a cylinder 722;

Fig. 23 shows a blended surface 750 according to the present invention that extends between the degenerate profile (point) 754 and the planar disk 758 having a circular curve 760 therein;

10 Fig. 24 illustrates the results of a blending technique of the present invention for blending a surface between semi-circular ribbons 784a and 784b, wherein the resulting surface 786 is blended between these two ribbons.

Fig. 25 shows a blended surface 808 according to the present invention whose points $p(u,v)$ are determined using a "forward algorithm", wherein points in parameter space 158 are themselves parameterized according to points in an additional parameter space 828, and wherein the points 830 of the additional parameter space are used to efficiently determine the distance-
15 like measurements to the pre-images (in parameter space 158) of the profiles 812 and 816 (in object space);

Fig. 26 is a flowchart showing the steps for computing an interpolating curve according to the present invention using a one-dimensional embodiment of the computational techniques novel to the present invention;

Fig. 27 shows a flowchart of the steps performed when constructing an approximation to an isocline boundary of an isocline ribbon, wherein the boundary is opposite the profile for the isocline ribbon;

20 Figs. 28A and 28B show a flowchart for a program that constructs a more precise isocline ribbon boundary than the approximation resulting from Fig. 27;

Figs. 29A - 29C illustrate a flowchart for modifying one or more subsurfaces S_i of a composite surface S_0 by changing a geometric characteristic of an isocline handle and/or a ribbon tangent for a marker on one or more profile curves defining the boundaries for the subsurfaces S_i ;

Figs. 30A and 30B provide a flowchart of a program invoked by the flowchart of Figs. 29 for deforming subsurfaces S_i in real time as a user modifies an isocline handle and/or ribbon tangent;

Fig. 31 is a flowchart of the high level steps performed by a user interacting with an embodiment of the present invention for changing the shape of a surface;

Fig. 32 pictorially illustrates examples of values for parameters used in the flowchart of Fig. 26 for computing an interpolating curve $C(u)$;

Fig. 33 shows four profile curves P_{11} , P_{12} , P_{21} and P_{22} wherein it is desired to generate a surface, bounded by these profiles and wherein the surface is defined by these four profiles (and their associated isocline ribbons);

Figs. 34 and 35 illustrate the intermediary surfaces generated during the performance of one method for creating a 4-sided patch (Fig. 36) from two 2-sided blends using the four profile curves of Fig. 33. That is, a blended surface S_1 (Fig. 34) is generated using the isocline ribbons R_{11} and R_{12} (for the profiles P_{11} and P_{12} , respectively), and a blended surface S_2 (Fig. 35) is generated using the isocline ribbons R_{21} and R_{22} (for the profiles P_{21} and P_{22} , respectively);

Fig. 36 shows a resulting blended surface S derived from S_1 (shown in Fig. 34), and S_2 (shown in Fig. 35), wherein S is generated according to Formula (11) provided in Section 5 of the detailed description hereinbelow;

Fig. 37 shows the geometric objects used in an embodiment of the present invention for generating a surface S from two surfaces S_L and S_R . In particular, this figure introduces the notational conventions subsequently used in Figs. 38 through 41;

Fig. 38 illustrates one embodiment of the present invention for generating a four-sided patch;

Fig. 39 illustrates an alternative embodiment of the present invention for generating the four-sided patch also generated in Fig. 38;

Fig. 40 shows the notational correspondences between the geometric objects of Fig. 38 and those of Fig. 39;

Fig. 41 shows a possible geometric configuration of Fig. 38, wherein the profiles P_3 and P_4 of Fig. 38 are degenerate;

Figs. 42A and 42B illustrate the movement of a marker 2002 that is constrained to reside on the profiles curves 2003 and 2004;

5 Fig. 43 illustrates constraints on composed profile curves and their corresponding isocline ribbons for providing tangent plane continuity between two blended surfaces S_1 and S_2 ;

Fig. 44 shows a profile P , associated isocline ribbons RL and RR , and various handles used in describing the conditions for achieving $G1$ continuity on P .

DETAILED DESCRIPTION

10 I. **Introduction**

Fig. 1 illustrates the use of an embodiment of the present invention for designing a surface 62 that interpolates any two parametric surfaces such as between the half cylinder surface 30 and the surface 34. That is, the surface 62 is generated via a novel surface interpolation process, wherein constraints on surface 62 shape are provided by the feature curves 54, 58 and 60, and their associated novel control geometry (e.g., isocline ribbons). In particular, the following constraints are satisfied by
15 the surface 62:

(a) one or more geometric characteristics of the surface 30 along the feature curve 54 are imposed on the surface 62,

(b) one or more geometric characteristics of the surface 34 along the feature curve 58 are imposed on the surface 62, and

20 (c) the surface 62 interpolates through the feature curve 60, wherein the surface 62 tangents along the extent of curve 60 are derived from (e.g., identical to) the isocline ribbons 61 and 63.

Thus, using the present invention, a designer can design a surface specified in terms of: (a) a relatively small number of carefully constructed and positioned feature curves, and (b) the desired slope(s) of the surface along the extent of these feature curves (via isocline ribbons). Moreover, using the present invention, such a designed surface not only interpolates fairly between the feature curves but also may obey other imposed constraints such as convexity, concavity, and/or predetermined curvature ranges.

5 Additionally, the present invention can be used to blend a surface region into an object being designed. For example, Fig. 2 illustrates the blending of a circular disk 66 into the cylindrical surface 30. Moreover, the present invention can also be used to construct bosses, dimples, logos, and embossing as well as to recursively design surfaces as one skilled in the art will come to appreciate from the disclosure herein.

 At least one embodiment of the present invention differs from traditional approaches to computer-aided design (CAD)
10 in that with the present invention, a desired geometric object (e.g., a surface) that may be created as a plurality of geometrically and computationally unrelated patches (e.g., three-, four-, five-sided bounded surfaces), which may be subsequently pieced together in a way that satisfies certain constraints at the boundaries between the patches. Thus, the desired geometric object can be designed by piecing together the plurality of unrelated geometric sub-objects (subsurfaces) in a manner that interpolates, blends and/or trims these sub-objects so that, across the boundaries and/or regions therebetween, constraints such as continuity,
15 differentiability, and/or curvature are satisfied. This is fundamentally different from the traditional approaches to CAD in that only four-sided NURBS, Bezier, Hermite, Coons, Gordon or Booleans of implicit surfaces are patched together in prior art systems.

2. Blending between Geometric Objects

 A fundamental geometric object design technique of the present invention is the blending between two parametric geometric objects such as surfaces and, more particularly, the manner in which such blending is performed. As defined in the
20 Definitions Section above, a *“parametric geometric object”* (e.g. a surface) may be defined as a result of a mapping from a (simple) coordinatized geometric object (parameter space) such as a bounded plane to another (typically, more complex) geometric object (object space). When the parameter space is a bounded plane, two coordinates or parameters (denoted u and

v) may by way of example be used to uniquely identify each point in the parameter space. When the object space is three-dimensional, for every (u,v) point in the bounded plane parameter space, a function may associate a point (x,y,z) in the object space.

By convention, a planar parameter space is usually assumed to be the unit square, which means that both u and v vary between 0 and 1, although it is within the scope of the present invention to utilize other parameter space geometries and coordinate ranges.

In one embodiment of the present invention, in order to blend between two parametric surfaces $S_1(u,v)$ and $S_2(u,v)$ each having the unit square as their parameter space, each surface S_1 and S_2 has associated therewith a respective blending function $B_1(u,v)$ and $B_2(u,v)$, wherein each of the blending functions has, for example, (0,1) as its range (as well as satisfying other properties given hereinbelow). Consequently, a new surface may be defined by the following formula:

$$S(u, v) = S_1(u, v) \cdot B_1(u, v) + S_2(u, v) \cdot B_2(u, v) \quad (1)$$

Note that the blending functions B_1 and B_2 are typically chosen so that the resulting blended surface S is the same as S_1 on a boundary with S_1 , and the same as S_2 on a boundary with S_2 . This is achieved by devising B_1 and B_2 so that $B_1=1$ and $B_2=0$ on the boundary with S_1 while having $B_2=1$ and $B_1=0$ on the boundary with S_2 . In Fig. 3, for example, if S_1 is the surface 30 and S_2 the strip 34, and one boundary is the vertical line 54 of the surface 30 and the other boundary is the curve 58 on the strip 34, then the surface 62a is S , which runs between these two boundaries and is tangent to S_1 and S_2 at the boundaries.

2.1. The Blending Functions

Blending functions may be provided for blending between geometric objects of various types. For example, blending functions for blending between two volume filling geometric objects can be provided. However, to simplify (and clearly illustrate) the novel blending process and the associated blending functions of the present invention, the discussion here is initially limited to blending between two curves, or blending between two surfaces. Accordingly, for two surfaces S_1 and S_2 to be blended together,

the blending functions $B_1(u,v)$ and $B_2(u,v)$, respectively, are appropriately set to either 0 or 1 on the boundaries of a blended surface generated by the present invention.

Referring to Fig. 4, wherein it is assumed that the boundaries 78a, 78b in parameter space correspond to the profiles 54 and 58 in object space, for any curve 80 on the blended surface such as surface 62a, there is a related pre-image (e.g., line 86) defined in parameter space as indicated. Note that for simplicity the boundaries 78a and 78b, and the pre-image of curve 80 are straight, but they need not be so.

Assuming (again for simplicity) the blending functions $B_1(u,v)$ and $B_2(u,v)$ have their domains in the unit square (as their parameter space), for any point (u, v) in this parameter space it is important to determine some measure of how "close" the point (u,v) is to the boundary curves (e.g., boundary curves 78a and 78b) and, more generally, to the pre-images of profile curves. Such closeness or distance-like measurements may be used in specifying the blending functions and/or their resulting values. Note that there are many ways to compute such a closeness or distance-like measurement in parameter space. For instance, if a boundary 78 (or profile pre-image) is a straight line, then such a parametric distance to a (u,v) point is easily calculated as the length of a perpendicular line segment to the boundary line through the point. Additional techniques for computing parametric distances are described hereinbelow (e.g., Sections 2.3 and 2.4).

Assuming parameter space is still the 2-dimensional space of (u,v) points, a blending function \tilde{B}_i (wherein $1 \leq i \leq N$ for some fixed N number of boundary curves) can be computed a function of a univariate distance-like function $\tilde{B}_i(D_i)$, where D_i is in turn a function of (u,v) so that $\tilde{B}_i(D_i) = \tilde{B}_i(D_i(u, v)) = B_i(u, v)$, wherein $D_i(u,v)$ is a distance-like function to the pre-image C_i^{-1} of a boundary curve C_i (in object space) of a surface S . Note that such distance-like functions must satisfy the condition that as (u,v) gets arbitrarily close to the i^{th} boundary curve pre-image C_i^{-1} (such as measured in conventional Euclidian distance), then $D_i(u,v)$ gets arbitrarily close to zero. Examples of such blending functions \tilde{B}_i and distance-like functions D_i are provided hereinbelow.

Since many of the most useful blending functions B_i are of the form $\tilde{B}_i(D_i)$, unless additional specificity is required, B_i will be used hereinbelow to denote both: (a) the blending function $B_i(u,v)$ initially discussed above, and (b) the blending function $\tilde{B}_i(D_i)$ for some distance-like function D_i . If, however, a clear distinction is required between the blending functions of (a) and (b), the domain of the blending function can be used to indicate which blending function is indicated. As an aside, note that Formula (1) applies equally well for the blending functions $\tilde{B}_i(D_i)$, $i = 1, 2$; that is,

$$S(u, v) = S_1(u, v) \cdot \tilde{B}_1(D_1) + S_2(u, v) \cdot \tilde{B}_2(D_2). \quad (2)$$

If a point (u,v) is close to the i^{th} pre-image boundary $i=1,2$, then $\tilde{B}_i(D_i)$ is expected to be small and the point is mapped (into object space) close to the i^{th} boundary.

A good collection of blending functions B_i not only allows the mapping, S , of a blended surface to be coincident with the desired perimeter (profile) curves, but will do so in a manner so that the resulting blended surface between two or more such perimeter curves of, e.g. for example, different initial surfaces will preserve such characteristics as the continuity of curvature with these initial two surfaces. That is, the blended surface “heels” to each of the initial surfaces. Also, it is preferred that the blending functions B_i allow the new surface to be fair. Figure 5 shows a graph of a pair of desirable blending functions for B_i , $i=1,2$.

For profile curves P_1, P_2 of two surfaces S_1 and S_2 , wherein a blended surface is desired between P_1 and P_2 , assume that the profiles P_1 and P_2 have parametric pre-images that correspond, respectively, to $u=0, u=1$ of the unit square $\{(u,v) \mid 0 \leq u \leq 1 \text{ and } 0 \leq v \leq 1\}$, then some useful properties for blending functions B_1 and B_2 are:

$$(1.1) \quad B_1=1 \text{ at } u=0 \text{ and } B_1=0 \text{ at } u=1. \quad B_2=0 \text{ at } u=0 \text{ and } B_2=1 \text{ at } u=1.$$

$$(1.2) \quad \text{The derivatives } B_1' \text{ and } B_2' \text{ equal } 0 \text{ wherever } D_i(u,v)=0 \text{ and } D_i(u,v)=1, i=1,2. \text{ This enforces smooth (tangent continuous) transitions between the blended surface } S \text{ and the initial surfaces } S_1 \text{ and } S_2. \text{ If higher}$$

order derivatives are also zero, then higher order continuity between surfaces can be realized, usually improving its fairness.

(1.3) $B_1 + B_2 = 1$ for all points (u,v) . This is called a "*partition of unity*," and it provides for the

generation of a *convex combination* of the surfaces S_1 and S_2 to which a new blended surface abuts. Note that this

5 tends to keep the new blended surface from drifting too far from the initial surfaces S_1 and S_2 .

There are numerous embodiments for defining blending functions. One useful embodiment is:

$$B_1(D_1) = \cos^2(D_1 \frac{\pi}{4}) \text{ and } B_2(D_2) = \sin^2(D_2 \frac{\pi}{4}) \quad (3)$$

which gives arbitrarily high order continuity of the blending functions, which is needed to achieve the same high order continuity

between the initial blended surfaces. Another alternative is to choose polynomial functions with the above properties (1.1)

through (1.3). For example, a quintic polynomial can be chosen with zero second derivative at $D=0$ and $D=1$, thereby providing

10 beneficial curvature characteristics (see Section 4.4).

In addition to the blending functions described hereinabove, the following are examples of additional blending functions:

$B_1(x)$ and $B_2(x)$ are polynomials satisfying the following constraints:

$$B_1(0) = 1, B'_1(0) = 0, B_1(1) = 0, \text{ and } B'_1(1) = 0$$

15 $B_2(x) = 1 - B_1(x)$

Note that additional constraints regarding high order derivatives (e.g., equal to 0 at $x=0$ and/or 1) may

also be imposed. For example, if $B''_i(0) = B''_i(1) = 0$, $i=1,2$, then C^2 continuity is attained with the objects from

which interpolating and/or blending is performed.

An example of polynomial blending functions satisfying these constraints is:

$$B_1(x) = (1-x)^2 + 5x(1-x)^4 + 10x^2(1-x)^3$$

$$B_2(x) = 1-B_1(x).$$

Note that $B_1(x)$ may be derived as a Bezier curve with six control points, P_1, \dots, P_6 , as shown in Fig. 6D.

Moreover, note that since

$$5 \quad B'_1(x) = -30x^2 \cdot (1-x)^2 \text{ and}$$

$$B''_2(x) = 60x \cdot (1-x)^2 - 60x^2 \cdot (1-x),$$

that

$$B'_1(1)=0, \quad B''(0)=0, \quad B''_1(1)=0, \quad B'_2(0)=0, \quad B'_2(1)=0, \quad B''_2(1)=0, \quad B'_1(0)=0 \text{ and}$$

$$B''_2(0)=0.$$

- 10 (c) Any composition of blending functions as described hereinabove with a bijective (e.g., one-to-one and onto) parameterization function $P:[0,1] \rightarrow [0,1]$ may be composed with a blending function to obtain another blending function. As a specific example, let $P(x) = 2c(x-x^2) + x^2$, where c is a constant "skew" factor, then a new blending function may be construed as $B(P(x))$. Thus, when $c = 1/2$, $P(x) = x$. Moreover, as c varies, the inflection point of the graph of $P(x)$ moves as shown in Figs. 6A-6C. Note that the blending
- 15 function of Fig. 6B (wherein $c > 1/2$) will cause the blending curve (and/or surface or other geometric object) to retain the geometric characteristics of the object used for blending that corresponds to the $x=0$ axis in the graph of Fig. 6B for a larger part of the surface.

To reduce the real-time design computational overhead incurred for evaluating blending functions, the values for the blending functions may, in one embodiment, be tabulated prior to a design session at a sufficiently high resolution and stored

20 in memory in a manner that allows efficient indexed access to a closest approximation to the actual blending function value.

2.2. Extending Blending to N-sided Regions

In one embodiment of the present invention, a novel general form for blending over a region that is bounded by each edge e_i of a parametric surface S_i ($i=1,2,\dots,N$ and $N\geq 2$) is the following weighted sum of points $S_i(u_i(p),v_i(p))$:

$$S(p) = \sum_{i=1}^N \left[\prod_{\substack{j=1 \\ j \neq i}}^N B_j(D_j(p)) \right] S_i(u_i(p), v_i(p)) \quad (4)$$

where:

- (a) p is a variable denoting points in a common parameter space for the surfaces S_i ;
- 5 (b) $D_i(p)$ is a distance-like measurement to the pre-image of the i^{th} edge e_i in the common parameter space;
- (c) B_i is a blending function which is zero when D_i is zero and monotonically increases as D_i increases; and
- (d) u_i and v_i are parameterization functions that transform p from the common parameter space to the (any) intermediate parameter space for S_i .

Note that by dividing by the sum of the products of the blending functions, B_i ,

$$(e.g., \sum_{i=1}^N \left[\prod_{\substack{j=1 \\ j \neq i}}^N B_j \right])$$

- 10 the formula (4) can be normalized with respect to the blending functions. Further note that Formula (4) resembles Formula (1) when $N=2$, and is in fact an extension thereof. That is, for $N=2$, B_1 of Formula (4) has the functional behavior of B_2 in Formula (1), and B_2 of Formula (4) has the functional behavior of B_1 in Formula (1). That is, there is a subscripting notational change between the two formulas.

- As an example of Formula (4), consider the three-sided region 300 shown (in object space) in Figure 14. Applying
- 15 Formula (4) to thereby generate a surface, S , for region 300, the following equation is obtained:

$$S(p) = B_2(v) B_3(w) S_1(u) + B_1(u) B_3(w) S_2(v) + B_1(u) B_2(v) S_3(w) \quad (5)$$

where u , v and w as parameterization functions are the barycentric coordinates of p as one skilled in the art will understand.

An alternative method to define a blended surface over N -sided ($N \geq 4$) regions is provided by first applying the two-sided approach based on Formula (1) using R_{11} and R_{12} of Fig. 34 as S_1 and S_2 , respectively in Formula (1) to thereby generate S_1 of Fig. 34. Additionally, Formula (1) is applied to the surfaces of Fig. 35, wherein S_1 and S_2 of Formula (1) are replaced by R_{21} and R_{22} respectively, to thereby generate S_2 of Fig. 35. The two resulting surfaces S_1 and S_2 of Figs. 34 and 35 respectively are, in turn, blended using Formula (2) wherein blending functions B_1 and B_2 are as described hereinabove, and the corresponding D_i are described hereinbelow. For example, given that each of the ribbons R_{11} , R_{12} , R_{21} , and R_{22} have a common pre-image, the D_i used in Formula (2) to compute distance-like measurements to the pre-images of the pair of edges P_{11} , P_{12} , P_{21} , and P_{22} (Figs. 34 and 35) may be:

(a) For a point P_1 of the (common) pre-image for S_1 of Fig. 34, $D_1(P_1) = \min(D(P_1, P_{11}), D(P_1, P_{12}))$ wherein D is the Euclidean distance between P_1 and the corresponding profile P_{1i} , and

(b) For a point P_2 of the (common) pre-image for S_2 of Fig. 35, $D_2(P_2) = \min(D(P_2, P_{21}), D(P_2, P_{22}))$.

Accordingly, the two surfaces S_1 and S_2 can be blended together using Formula (2) to obtain surface S of Fig. 36.

In another embodiment that is particularly useful for generating a four-sided blended patch, assume the following restricted but versatile scheme for defining profiles and ribbons:

(a) All handles are piecewise linear segments; and

(b) All blending is done with the functions $B_1(x)$ and $B_2(x)$ of Formulas (3.1).

Moreover, referring first to Fig. 37 in describing the present patch generation technique, the following labeling scheme is used.

For the profile, P :

m_l, m_r : the left and right hand markers, respectively, of the profile, P ;

h_l, h_r : the left and right hand profile handles, respectively, of the profile, P ;

s_l, s_r : the left and right hand isocline handles, respectively, of the profile, P ;

b_L, b_R : the left and right hand ribbon tangents at the respective left and right end points of isocline boundary R (these ribbon tangents also being denoted as "boundary handles").

Using the notation of Fig. 37, surfaces S_L and S_R may be defined, wherein S_L is bounded by the line segments corresponding to: s_L, h_L, b_L , and $d_L = (s_L + b_L) - h_L$, and S_R is bounded by the line segments corresponding to: s_R, h_R, b_R , and $d_R = (s_R + b_R) - h_R$. In particular, S_L and S_R are known in the art as "twisted flats," and accordingly, S_L is denoted as the left twisted flat, and S_R is denoted as the right twisted flat. Moreover, these surfaces may be evaluated using the following formulas (5.01a) and (5.01b):

$$S_L(u, v) = (1 - v, v) \begin{pmatrix} m_L & h_L \\ s_L & b_L \end{pmatrix} \begin{pmatrix} 1 - u \\ u \end{pmatrix} \quad (5.01a)$$

wherein the parameters u and v increase in transverse directions as illustrated by the u -direction arrow and the v -direction arrow (Fig. 37).

$$S_R(u, v) = (1 - v, v) \begin{pmatrix} h_R & m_R \\ b_R & s_R \end{pmatrix} \begin{pmatrix} 1 - u \\ u \end{pmatrix} \quad (5.01b)$$

wherein the parameters u and v also increase in transverse direction, with the u -direction being the reverse direction of the u -direction arrow of Fig. 37.

Accordingly, the isocline ribbon surface S (Fig. 37) can now be defined as follows:

$$S(u, v) = B_2(u) S_L(u, v) + B_1(u) S_R(u, v) \quad (5.02)$$

where conveniently, the u parameter is also the distance measure needed for B_1 and B_2 of Formulas (3.1). Thus, when $v = 0$, $S(u,0)$ is the profile; i.e., a blend between the control handles (h_L-m_L) and (h_R-m_R) . Additionally, note that when $v = 1$, $S(u,1)$ is the ribbon boundary R derived as a blend of vectors (b_L-s_L) and (b_R-s_R) . Also note that if b_L and b_R are translates of h_L and h_R , respectively, along s_L-m_L and s_R-m_R , respectively, then R is a translation of P , and such similarities may simplify the data storage requirements of the present invention.

For a plurality of isocline ribbons S_1, S_2, \dots, S_N , wherein each S_i is generated by Formula (5.02), such ribbons may now be used in the more general N -sided surface form below, which is a variation of Formula (4).

$$S(s, t) = \frac{\sum_{i=1}^N \left(\prod_{\substack{j=1 \\ j \neq i}}^N B_j(D_j(s, t)) \right) S_i(u_i(s, t), v_i(s, t))}{\sum_{i=1}^N \left(\prod_{\substack{j=1 \\ j \neq i}}^N B_j(D_j(s, t)) \right)} \quad (5.03)$$

Note that $D_i(s,t)$, $u_i(s,t)$ and $v_i(s,t)$ must be defined for this formula, i.e., the distance measure and the mappings from the general N -side patch parameter space (in s and t) to the parameter space of the ribbons S_i (in u and v).

For specific cases where $N = 2, 3, 4$ and $N \geq 5$ using the blended ribbons S_i , notice first that Formula (5.02) for the ribbon is a special case of Formula (5.03). For example, in Formula (5.02) the denominator is 1, the distance measure is just the u -parameter, and u and v correspond exactly to s and t . The formula for a two-sided surface is similar, except that the base surfaces are ribbons derived according to Formula (5.02) (denoted herein also a "twisted ribbons"); thus,

$$S(u, v) = B_2(v) S_1(u, v) + B_1(v) S_2(u, v) \quad (5.04)$$

which is similar to Formula (5.02), wherein the parameter u measures distance. It varies along the direction of the profile curve.

However, in Formula (5.04), the parameter v measures distance.

Referring to Fig. 38, wherein the isocline ribbons S_1 and S_2 are parameterized as indicated by the u and v direction arrows on each of these ribbons, these ribbons may be used to generate a four-sided patch. The two profiles P_1 and P_2 that vary in u are blended using the twisted ribbons S_1 and S_2 . The other two sides P_3 and P_4 are blended profiles derived from the isocline handles; that is, P_3 is a blend (e.g., using Formula (I)) of h_R^1 and h_L^1 , wherein h_R^1 is S_1 and h_L^1 is S_2 in Formula (I), and similarly, P_4 is a blend of h_R^2 and h_L^2 .

Note that the blended surface, S , of Fig. 38 has tensor product form. This can be shown by decomposing Formula (5.04) into a tensor form, wherein each of the ribbons S_1 and S_2 is derived from the Formula (5.01a) and (5.01b). That is, S_1 is a blend of S_L^1 and S_R^1 (Fig. 38) and S_2 is a blend of S_R^1 and S_R^2 . Accordingly, the decomposition is as follows:

$$\begin{aligned} S(u, v) &= (B_2(v) \ B_1(v)) \begin{pmatrix} S_1(u, v) \\ S_2(u, v) \end{pmatrix} \\ &= (B_2(v) \ B_1(v)) \begin{pmatrix} S_L^1(u, v) & S_R^1(u, v) \\ S_L^2(u, v) & S_R^2(u, v) \end{pmatrix} \begin{pmatrix} B_2(u) \\ B_1(u) \end{pmatrix} \quad (5.05) \\ &= (B_2(v) \ S_L^1 + B_1(v) \ S_L^2 \ B_2(v) \ S_R^1 + B_1(v) \ S_R^2) \begin{pmatrix} B_2(u) \\ B_1(u) \end{pmatrix} \end{aligned}$$

Thus, the last expression above shows that the same surface S can be generated by first creating the twisted ribbons in the v parameterization, and then second, blending in u . However, since the roles of u and v are symmetric, the twisted ribbons may be generated along the u parameterization and subsequently, the blending may be performed in v . That is, using the surfaces

S_L^3 and S_R^3 , S_L^4 and S_R^4 of Fig. 39 gives the same surface S as in Fig. 38. Thus, in either technique for deriving S , the inputs are the same; that is, m_L^i , m_R^i , h_L^i , h_R^i , s_L^i , s_R^i , b_L^i , and b_R^i , where "i" denotes the profile P_i ($i = 1, 2, 3, 4$) to which the inputs apply. Note that the correspondences between the various inputs is shown in Fig. 40.

So, overall, the two-sided patch of Formula (5.04) provides a very versatile four-sided patch. Moreover, its evaluation is also efficient. Thus, by expanding the S_L^i and S_R^i of Formula (5.05) using Formulas (5.01a) and (5.01b), the following expression may be obtained:

$$(B_2(v), B_1(v)) \left[(1-v, v) \begin{pmatrix} \begin{pmatrix} 1 \\ L \end{pmatrix} & \begin{pmatrix} 1 \\ R \end{pmatrix} \\ \begin{pmatrix} 2 \\ L \end{pmatrix} & \begin{pmatrix} 2 \\ R \end{pmatrix} \end{pmatrix} \right] \begin{pmatrix} 1-u \\ u \end{pmatrix} \begin{pmatrix} B_2(u) \\ B_1(u) \end{pmatrix} \quad (5.06)$$

where $\begin{pmatrix} i \\ L \end{pmatrix}$ and $\begin{pmatrix} i \\ R \end{pmatrix}$ are the appropriate matrices from Formulas (5.01a) and (5.01b). Note that when evaluating an instantiation of this expression, the B_i should probably be table driven.

The above formulation is mathematically sound, but to use it in a geometrically intuitive fashion still requires judgment on the user's part. Thus, in certain degenerate cases, some mathematical aids are also in order. A common instance is where two of the profiles (e.g., P_1 and P_2) intersect each other, as in Fig. 41. This is a degenerate case since profiles P_3 and P_4 (of Fig. 38) are zero length, and share end markers (i.e., $m_L^3 = m_L^4$ and $m_R^3 = m_R^4$).

Note, however, that Formula (5.04) still defines a surface S , but it is easy to see that the surface may loop at the profile intersections. To eliminate this looping and still maintain handle-like control at the markers, the twisted ribbon of Formula (5.04) may be sealed by a function of u . One function that is 1 at $u = 1/2$ and 0 at $u=0$, is:

$$\alpha(u) = 1 - 4(u - \frac{1}{2})^2 \quad (5.07)$$

Thus, Formula (5.01) is adjusted to be:

$$S(u, v) = B_2(v) \alpha(u) S_1(u, v) + B_1(v) \alpha(u) S_2(u, v) \quad (5.08)$$

Such a function (5.08) will likely remove most loops.

The ability to diminish the ribbon at the ends suggests other applications. A scaling function such as

$$\alpha_1(u) = 1 - u^2 \quad (5.09)$$

diminishes the ribbon at the $u=1$ end, while

$$\alpha_1(u) = 1 - (u - 1)^2 \quad (5.091)$$

- 5 diminishes it at the $u = 0$ end. This is an effective way to make a triangular (three-sided) surface, as one skilled in the art will understand.

2.2.1 Bosses and Dimples from 2-Edges

- The so-called "boss" feature may be obtained from a blending between two profile edges. The profiles may be provided as, for example, semicircles 780a and 780b of Fig. 24 having isocline ribbons 784a and 784b, respectively. The ribbons 784a and 10 784b are in distinct parallel planes. When these ribbons are blended together, a surface 786 (Fig. 24) is obtained which may be considered a boss or a dimple. Note that many variations, i.e., domes, rocket tips, mesas, apple tops, etc. may be generated similarly. Moreover, if the top semicircular ribbon is rotated, the boss can be made to twist. This scheme can be used to transition between tubes, like a joint, as one skilled in the art will understand.

- Note that in another embodiment, blending may be performed by using a neighborhood about each boundary curve 15 (in object space) as a default isocline ribbon from which to blend using Formula (1) or Formula (4). Thus, by defining a value $\epsilon > 0$, and taking a strip and width of each surface along the boundary to which the surface is to be blended, these strips may

be used as isocline ribbons. Accordingly, the surface boundaries become profile curves and pre-images thereof may be used in the Formula (1) or Formula (4).

2.3. Profile Curves

Since the present invention can take a few well-positioned (object space) profile curves of various types and generate
5 a corresponding surface therethrough, as a blended surface according to Formula (1) above, there are two parameter space pre-image curves for each of the surfaces S_1 and S_2 wherein these curves are boundaries for the blending functions B_1 and B_2 ; that is, a curve at $D_i=0$ and at $D_i=1$ for each blending function B_i . In fact, there may be eight curves, as illustrated in Fig. 7, that may be used to define a blended surface. That is, there may be two curves 78a and 78b in the parameter space of S_1 and two additional curves 78c and 78d in the parameter space for S_2 (of course, in many cases these two parameter spaces are identified).
10 Additionally, there are the mappings of the curves 78 to the two surfaces 30 and 34, thereby providing the corresponding image curves 90, 54, 58 and 91, these having respective pre-images 78a, 78b, 78c and 78d.

Note that in the case where S_1 and S_2 have identical parameter spaces, profile 78b is the pre-image of the profile 54. Moreover, if S_2 of 78d (= 78b) is profile 58, then 78b is included in the pre-image of each of S_1 , S_2 and blended surface 62.

When the present invention is used for surface design, a user or designer may think of designing a blended surface by
15 continuously pulling or deforming one profile curve of an initial surface to thereby create a new surface between this initial surface and a profile curve of another initial surface.

Note that different types of profile or boundary curves may be used with the present invention. In some embodiments of the present invention, such a profile curve, C , may typically have a parametric pre-image in a parameter space, i.e. $C^1(s) = (u(s), v(s))$ where s is a parameterization of the pre-image (e.g., $0 \leq s \leq 1$). Note that parametric curves such as C
20 include curves having the following forms: (a) conics including lines, parabolas, circles and ellipses; Bezier, Hermite and non-uniform rational b-splines (NURBS); (b) trigonometric and exponential forms; and (c) degenerate forms like points. Additionally,

note that these curve forms may be categorized orthogonally by other characteristics such as open, closed, degenerate and composite, as one skilled in the art will understand.

Profile curves include curves from the following curve-type categories (2.3.1) through (2.3.5).

2.3.1. Open Curves

5 An “*open curve*” is one in which the end points of the curve are not constrained to be coincident; e.g., the end points may be freely positioned. Open curves are probably the most common type used by the present invention when defining an arbitrary collection of curves (profiles) for generating a surface (in object space), wherein the surface is constrained to pass through the collection of curves.

2.3.2. Closed Curves

10 When a curve’s end points match, the curve is denoted as “*closed*.” This means that the beginning point of the curve is the same as the ending point of the curve. Closed curves delimit regions of, e.g., a surface, and are especially useful for setting special design areas apart. One example of this is the *label surface* for containers (described in the Definitions Section hereinabove); e.g., surface 66 of Fig. 2. That is, a label surface is a region that must be of a particular surface type, denoted a developable surface, so that a label applied thereto will not crease or tear. Each such label surface is highly constrained and is
15 usually separated from the rest of the design by a closed curve (such a curve can also serve aesthetic purposes in the design of the container). Fig. 8 shows an elliptic region 100 blended into a cylinder 108, wherein the closed curve 110 delimits the elliptic region. A closed curve may often match tangencies at end points.

2.3.3. Degenerates

 Several ways exist to generate a degenerate profile. In one technique, an open curve may be of zero length, or a closed
20 curve may enclose a region of no area. In such cases, the result is a point that may blend with an adjacent surface. Fig. 9 shows a point blend created from blending between a degenerate circular disk (i.e., the point labeled S_1) and the cylinder 116 (also denoted as S_2). Accordingly a simple boss 112 is created on the cylinder 116. In particular, for appropriate blending functions

B_i , $i=1,2$, a blended surface between S_1 and S_2 can be obtained using Formula (1). Moreover, since Formula (4) can be used instead of Formula (1), a surface can be generated that blends between a plurality of points (i.e., degenerate profiles) and an adjacent surface. Figs. 22 and 23 show additional blends to degenerate profiles.

Fig. 22 shows a blended surface 710 that extends between the degenerate profile (point) 714, and the circular end 718 of the cylinder 722. In particular, the blended surface 710 is a blending of the isocline ribbons 726 and 730, wherein the isocline ribbon 726 is a planar disk having the degenerate profile 714 as its center point, and the isocline ribbon 730 has the circular end 718 as its profile. Thus, letting S_1 be the isocline ribbon 726, and S_2 be the isocline ribbon 730 in Formula (1), the distance-like measurements (in their corresponding parameter spaces) can be equated to:

- (a) the radial distance from the degenerate profile 714 on the isocline ribbon 726;
- (b) the distance away from the profile 718 on the isocline ribbon 730.

Fig. 23 shows another blended surface 750 that extends between the degenerate profile (point) 754, and the planar annulus 758 having a circular curve 760 therein (and having, optionally, a central hole 762 therethrough with curve 760 as its boundary). In particular, the blended surface 750 is a blending of the isocline ribbon 766 (for the degenerate profile 754), and the annulus 758 (which, e.g., can optionally be an isocline ribbon to the surface 770 wherein curve 760 is a profile). Thus, letting S_1 be the isocline ribbon 766 and S_2 be the annulus 758, the distance-like measurements (in their corresponding parameter spaces) can be equated to:

- (a) the radial distance from the degenerate profile 754 on the isocline ribbon 766;
- (b) the distance away from the curve 760 on the annulus 758.

2.3.4. Composite Curves

The novel geometric design techniques of the present invention can also be utilized with composite curves. Composite curves are general curve forms that include other curves as sub-curves, wherein the sub-curves may cross or may kink, e.g., at endpoints. In utilizing composite curves as, e.g., profiles, the definition of a distance-like measurement for a composite curve

is important. Fig. 10 shows a composite curve 120 that includes two crossing sub-curves 124 and 128. However, such composite curves can also have their sub-curves strung end-to-end.

Assuming the sub-curves $C_j, j=1,2,...,N$ of a composite curve C are parameterized and have a common parameter space, a distance formula (in parameter space) for determining a distance-like measurement D to the pre-images of the sub-curves C_j

5 is:

$$D(p) = D_N(P), \text{ and}$$

$$D_k(P) = d_k(P) + D_{k-1}(P) - [d_k^2(P) + D_{k-1}^2(P)]^{\frac{1}{2}} \quad (5.5)$$

where $k=2,...,N$ and $D_1(p) = d_1(P) =$ a distance measurement between P and C_1 , and $D_k(P) =$ a distance measurement between P and C_k . Thus, $D(p)$ can be used as the input to a blending function, $B(D)$, for blending one or more surfaces to the composite curve, C .

2.3.5. Trimming Curve

10 The present invention allows a surface to be “trimmed,” wherein trimming refers to a process for constraining or delimiting a surface to one side of a particular boundary curve (also denoted a trim curve). In particular, for parameterized surfaces, the pre-image of a trim curve, e.g., in the (u,v) parameter space of the surface, identifies the extent of the pre-image of the surface to remain after a trimming operation. A trim curve may be a profile curve, and the desired trimmed surface is that part of the original untrimmed surface that typically lies on only one side of the trim curve. An example is shown in Fig. 11,

15 wherein the original untrimmed surface is the generally rectangular portion 130. The rounded surface 134 is a “label” surface that is trimmed to the curve 138 from the original surface 130. Note the trim profile 138 may have an associated isocline ribbon (not shown) for one or more adjacent surfaces (e.g., surface 142) that heel to an isocline ribbon at the trimming profile 138. The use of isoclines for modifying the shape of such adjacent surfaces is an important technique in creating a smooth transition from the adjacent surfaces to a trimmed surface.

Note that the present invention may include a trimming technique to create a hole in a geometric object. By extruding a depression in a front surface of the geometric object through a back surface of the object, and then trimming the front surface to exclude the corresponding portion on the back surface, a hole can be constructed that can be used, e.g., as a handle of a container.

5 2.4. Distance Metrics

Some techniques for computing distance-like measurements have already been provided hereinabove. In this section, additional such techniques are described. The efficiency in computing how close a point in parameter space is to one or more particular geometric object pre-images (curves) in parameter space can substantially impact the performance of a geometric design and modeling embodiment of the present invention. In general, for computing such distance-like measurements (these
10 being, in general, a monotonic function of the conventional Euclidean distance metric) in parameter space between points and curves, there is a trade-off between the complexity of the curve and how efficiently such measurements can be evaluated. In general, the simpler the curve, the faster such distances can be determined. As an aside, it should be noted that for a parameter space curve and its image curve (in object space), these curves need not be of the same computational type (e.g., polynomial, transcendental, open, closed, etc.). Indeed, a parameter space curve may be quite simple and still be the pre-image of a
15 complicated surface curve in object space. For example, the parameter space curve corresponding to the Bezier curve 58 in Fig. 1 may be a straight line. By keeping the parameter space curve as simple as possible, fast distance computations are possible.

2.4.1. Parametric Distance Calculations for Blending

This section describes a variety of methods for calculating a distance-like measurement (more generally, a monotonic function of the conventional Euclidean distance metric) to a number of candidate parameter space curves, wherein the methods
20 are listed in a roughly increasing order of computational complexity.

Assume a blended surface is to be generated between two profile curves P_1 and P_2 , each having isocline ribbons, wherein each ribbon is parametric and has, e.g., the planar unit square $[0,1] \times [0,1]$ as the common parameter space for the ribbons. One

distance-like function capable of being used for blending is a function that is dependent on only one or the other coordinate of points represented by the coordinate pairs (u,v) in the common parameter space. That is, assuming the profile curves P_1 and P_2 of the isocline ribbons are such that their pre-images are the vertical lines $u=k_1$ and $u=k_2$ for $0 \leq k_1 \leq k_2 \leq 1$, then the corresponding distance-like functions can be $D_1(u,v) = (u-k_2)/(k_1-k_2)$ and $D_2(u,v) = (u-k_1)/(k_2-k_1)$. Moreover, if the pre-images are the parameter space bounding vertical lines $u=0$ and $u=1$ (i.e., $k_1=0$ and $k_2=1$), then the corresponding distance-like function can be $D_1(u,v) = 1-u$ and $D_2(u,v) = u$, and accordingly such simple distance-like functions can be computed very efficiently.

In order to maintain the desired simplicity in parametric distance computations when there are pre-images to more than two profiles for blending therebetween, three methods can be employed for computing parametric distance-like measurements. Each of the three methods is now described.

A triangular domain in parameter space bounded by, e.g., three profile curve pre-images (that are also curves) can be parameterized with respect to the vertices v_1, v_2 and v_3 of the triangular domain using three (real valued) parameters r, s and t with the additional constraint that $r+s+t=1$. In other words, a point p in the triangular domain having the vertices v_1, v_2 and v_3 can be represented as $p = r*v_1 + s*v_2 + t*v_3$. The r, s, t parameters are called “*barycentric coordinates*” and are used for three-sided surfaces such as the surface 300 of Fig. 14 in parameter space.

Domains in parameter space that are bounded by the pre-images of four profiles (denoted the four-sided case) can be a simple extension of the domain having bounds on two opposing sides (denoted the two-sided case). In the two-sided case, if parameterized properly, only one parameter, u , need be used in the distance-like function computation. In the four-sided case, both parameters u and v may be employed, as well as their complements (assuming an appropriate representation such as the unit square in parameter space). Thus the distance to the four profile pre-image boundaries in parameter space can be $u, v, 1-u$, and $1-v$ (i.e., assuming the pre-images of the profiles are $u=0, v=0, v=1, u=1$).

To determine barycentric coordinates for parametric space domains, assuming the pre-images of the profiles are line segments that form a polygon, the approach illustrated in Fig. 12 (illustrated for a five-sided polygon 148 having vertices v_1, v_2, v_3, v_4 and v_5) may be utilized, wherein the profile pre-images are the heavy lines labeled 149a through 149e. To determine a distance-like function, first, stellate, i.e., make a star from, the pre-image polygon 148 by extending each of the sides 149a through 149e of the polygon until they intersect with another extended side having a side 149 therebetween. Thus, the intersection points 150a through 150e are determined in the five-sided case of Fig. 12. Subsequently, the line segments 152a through 152e from the corresponding points 150a through 150e to a point p in the polygon may be constructed. The resulting distance-like measurements are the lengths of the line segments 153a through 153e from p to the sides 149a through 149e of the polygon 148. Accordingly, the distance from p to the i^{th} side 149 ($i = a, b, c, d, e$) of the polygon 148 is the distance along the i^{th} line segment 153 from p to the boundary edge of the polygon 148. Note that by dividing each resulting distance-like measurement by the sum of all the distance-like measurements to the point p , the distance-like measurements can be normalized.

2.4.2. Straight Line

A straight line is represented by the equation, $au + bv = c$, wherein a, b, c are constants. A convenient (unsigned) distance to a line is obtained by

$$D(u, v) = |(a, b) \cdot (u, v) - c|. \quad (6)$$

For a more intuitive version that corresponds to Euclidean distance, Formula (6) can be normalized to obtain

$$D(u, v) = |(a, b) \cdot (u, v) - c| / (a^2 + b^2)^{1/2}, \quad (7)$$

by dividing by the length of the gradient.

2.4.3. Conics

Conics include parabolas, hyperbolas and ellipses. The general form of a conic is

$$Au^2 + Buv + Cv^2 + Du + Ev + F = 0.$$

Its unsigned distance can be computed by

$$D(u, v) = \left| (u, v) \cdot \begin{bmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{bmatrix} \cdot \begin{pmatrix} u \\ v \end{pmatrix} + \left(\frac{D}{E} \right) \cdot (u, v) + F \right| \quad (8)$$

This can also be normalized by dividing through by the length of the gradient of the function to make a more suitable distance-like function, which is Euclidean in the case of the circle. Note that Farin in Introduction to Curves and Surfaces, Academic Press, 4th ed., 1996, (incorporated herein by reference) gives the conversion between the implicit form above and a rational parametric form. Thus, Formula (8) can be used regardless of whether the conic is represented implicitly or parametrically.

2.4.4. Polynomial Curves, both Parametric and Implicit

Assume that a parametric curve has been converted to a Bezier form as, for example, is described in the Farin reference cited hereinabove. Vaishnav in Blending Parametric Objects by Implicit Techniques, Proc. ACM Solid Modeling Conf., May 1993 (incorporated herein by reference) gives a method to change a curve from a parametric curve to an implicit curve numerically, wherein distance is implicitly measured in object space by offsetting the curve in a given direction that is based on some heuristics about how the offset is to be computed. The value of the offset distance that forces the offset to go through the point is the distance measurement for that point. In particular, for a Bezier curve, this distance-like measurement may be worthwhile in that it is robust (i.e., not ill-conditioned) and reasonably fast to evaluate, requiring only two or three Newton-Raphson iterations on average, as one skilled in the art will understand. While this may be an order of magnitude slower than computing a distance measurement of a conic representation, it is much faster than the traditional method of computing a perpendicular distance, which is also unstable.

2.4.5. Piecewise Parametric Curves

The present invention also includes a novel technique for computing a distance-like measurement on complex curves in parameter space.

Referring to Fig. 13, assume that both boundary curves 156a and 156b are in the unit square parametric space 158 and are piecewise parametric polynomial curves that have corresponding sub-curves 160a, 160b of the same degree n . By connecting end points of the corresponding sub-curves with line segments 164 (i.e., degree one curves), degree n by 1 Bezier patches 168 can be constructed in the unit square representation of parameter space 158. Note that each patch 168 can be considered as a second parameter space unto itself having coordinates (s,t) wherein:

- (a) for two Bezier sub-curves 160a and 160b (denoted herein $b_1(t)$ and $b_2(t)$, $0 \leq t \leq 1$), each value, t_0 , of t corresponds to a line segment, L_{t_0} , between $b_1(t_0)$ and $b_2(t_0)$, and
- (b) the L_{t_0} line segment is parameterized by s so that $L_{t_0}(s) \in b_1(t_0)$ when $s=0$ and $L_{t_0}(s) \in b_2(t_0)$ when $s=1$, wherein s varies proportionally with the distance between $b_1(t_0)$ and $b_2(t_0)$ when $0 < s < 1$. Accordingly, if the distance-like measurement between the curves $b_1(t)$ and $b_2(t)$ (and/or patch bounding line segments 164) is computed in the second parameter space, then for any (u,v) point interior to the patch, it is necessary to find the corresponding (s,t) point relative to the boundary curves of such a patch that can then be evaluated for determining the distance-like measurement. Since s is the linear parameter (corresponding to the distance of a point between the two corresponding sub-curves 160a and 160b that are joined at their endpoints by the same two segments 164), simple functions $f_1(s)$ and $f_2(s)$, such as $f_1(s)=s$ and $f_2(s)=1-s$, can serve as distance functions to $b_1(t)$ and $b_2(t)$, respectively. Note that the parameters u and v can both be represented as Bezier functions of s and t . In particular, to convert from (s,t) coordinates to (u,v) parameter space coordinates, a Newton type algorithm may be used, as one skilled in the art will understand.

Another approach for determining the distance-like measurement, in some circumstances, is to evaluate such patches 168 with a “forward algorithm.” That is, referring to Fig. 25, an object space blended surface 808 that blends between, e.g., profiles 812 and 816 (having isocline ribbons 820 and 824, respectively, to which the surface 808 heels) is shown. The profile 812 has as its pre-image curve 160a (in parameter space 158), and the profile 816 has as its pre-image curve 160b (in parameter space 158), wherein the portion of parameter space 158 for surface 808 is the patch 168. An additional parameter space 828 in s and t can be considered as a pre-image parameter space for the parameter space 158 wherein the pre-image of curve 160a is the vertical line segment at $s=0$, and the pre-image of curve 160b is the vertical line segment at $s=1$. If a sufficiently dense set of points 830 denoted by “x”s in the additional parameter space 828 is used to evaluate points (u,v) in patch 168 (e.g., by determining a closest point 830), then the corresponding points $p(u,v)$ on a blended surface 808 can be efficiently computed since the distance-like functions to pre-image curves 160a and 160b can be $D_1(u(s,t),v(s,t))=s$ and $D_2(u(s,t),v(s,t))=1-s$, respectively. This approach will generate the blended surface easily and quickly. Note, if the surface 808 does not require a subsequent trimming operation, this method is particularly attractive.

3. Blending Programs

Figure 17 shows a block diagram of the typical flow of design construction operations performed by a user of the present invention. Thus, profile handles may be needed to construct an associated profile, and the profile is required to construct the associated isocline ribbon, and the isocline ribbon may be required to obtain the desired shape of the associated object (e.g., a surface), which, in turn, is required to construct the desired geometric model.

Figs. 26 through 30 provide a high level description of the processing performed by an embodiment of the present invention that enables the novel real-time manipulation of the shape of geometric object representations so that a user can more efficiently and directly express his/her design intent. Moreover, it should be noted that a fundamental tenet of the present invention is a paradigm shift away from typical CAD systems. That is, in a typical CAD system the user must supply input for changing or modifying a geometric object and subsequently request processing of the input to commence. Thus, the user receives

feedback about his/her design at discrete user requested times. Instead, with the present invention, updates may be processed in real-time immediately upon input without the user explicitly indicating that update processing is to be performed. Accordingly, a user of the present invention can efficiently perform iterative approximations to a geometric object being designed without requiring the user to precisely calculate geometric characteristics for substantially all portions of the object. In particular, this can have substantial efficiency benefits in that for many geometrically designed objects (including machined parts), substantial portions of such objects may be satisfactorily designed using a wide range of geometrically shaped objects. Accordingly, the present invention allows many of these geometric objects to be designed without the user having to needlessly specify precision in those portions of the object where the precision is unnecessary.

In Fig. 26, the steps are shown for computing an interpolating curve according to the present invention using a one-dimensional variation of Formula (1) discussed in Section 2 hereinabove. Accordingly, in step 1004, the end points and tangents at the end points for the interpolating curve, $C(u)$, to be generated are obtained. In particular, the end points of this curve are assigned to the variables PT1 and PT2. Additionally, direction vectors for the interpolating curve $C(u)$ at the points PT1 and PT2 are assigned to the variables TAN1 and TAN2, respectively. Note that PT1, PT2, TAN1 and TAN2 can be supplied in a variety of ways. For example, one or more of these variables can have values assigned by a user and/or one or more may be derived from other geometric object representations available to the user (e.g., another curve, surface or solid representation). In particular, the direction tangent vectors denoted by TAN1 and TAN2 may be determined automatically according to a parameterization of a geometric object (e.g., a surface) upon which the points PT1 and PT2 reside.

In steps 1008 and 1012, the blending functions B_1 and B_2 are selected as discussed in Section (2.1) hereinabove. Note, however, that the blending functions provided may be defaulted to a particular pair of blending functions so that a user may not need to explicitly specify them. However, it is also within the scope of the present invention that such blending functions may be specifically selected by the user. In this regard, note that since the present invention is intended to express the user's geometric design intent, there may be a wide variety of blending functions that are acceptable since typically a user's intent is often

adequately expressed without the user purposefully determining highly precise input. That is, it is believed that a wide variety of blending functions may be acceptable for iterative approximation of a final geometric design since progressively finer detail can be provided by generating and/or modifying progressively smaller portions of the geometric object being designed using substantially the same blending functions. Said another way, since the present invention supports both the entering of precise (geometric or otherwise) constraints as well as the iterative expression of the user's intent at progressively higher magnifications, the high precision and/or small scale design features may be incorporated into a user design only where necessary.

In step 1016, the interpolating curve, $C(u)$, is computed using a variation of Formula (2) applied to a one-dimensional parameter space. An example of an interpolating curve, $C(u)$, with points $PT1$, $PT2$, and vectors $TAN1$ and $TAN2$ identified, is illustrated in Fig. 32.

In Fig. 27, a flowchart is provided showing the steps performed when constructing an approximation to an isocline boundary $R(u)$ for an object (e.g., a surface) S , wherein the points $PT1$ and $PT2$ delimit a profile curve corresponding to the isocline ribbon boundary approximation to be generated. In particular, the approximate isocline ribbon boundary generated by this flowchart is intended to approximately satisfy the isocline ribbon boundary definition in the Definitions Section hereinabove. More precisely, the isocline ribbon boundary approximation determined by the present flowchart will tend to match the isocline ribbon boundary definition for a portion of the object S between $PT1$ and $PT2$ depending on, e.g., how smooth the object is along the profile curve generated between $PT1$ and $PT2$. That is, the smoother (reduced curvature fluctuations), the more likely the match. Accordingly, in step 1104 of Fig. 27, the curve interpolation program represented in Fig. 26 is invoked with $PT1$, $PT2$ and their respective tangents $TAN1$ and $TAN2$ for the object (surface) S . Thus, an interpolating curve, $C(u)$, is returned that is an approximation to the contour of S adjacent to this curve.

In steps 1108 and 1112, a tangent (i.e., a picket) along the parameterization of the object S at each of the points $PT1$ and $PT2$ is determined, and assigned to the variables $PICKET1$ and $PICKET2$, respectively. Note that typically the pickets, $PICKET1$ and $PICKET2$, will be transverse to the vectors $TAN1$ and $TAN2$, although this need not be so. Subsequently, in steps 1116 and 1120,

the isocline ribbon points corresponding to PT1 and PT2 are determined and assigned to the variables, RIBBON_PT1 and RIBBON_PT2, respectively. Then, in step 1124, the curve interpolation program of Fig. 26 is again invoked with the values RIBBON_PT1, RIBBON_PT2, TAN1 and TAN2 to thereby generate the isocline ribbon boundary approximation, R(u). It is worthwhile to note that in some cases, the isocline ribbon approximation bounded by the interpolating (profile) curve C(u), the corresponding pickets (PICKET1 and PICKET2), and the newly generated isocline boundary R(u) does not necessarily form a surface. In fact, the curves, C(u) and R(u) may be substantially coincident (e.g., if PICKET1 is identical to TAN1, and PICKET2 is identical to TAN2).

In Figs. 28A and 28B, a flowchart for a program is provided for constructing a more precise isocline ribbon boundary than the approximation resulting from Fig. 27. In particular, in the flowchart of Figs. 28A and 28B, the program of Fig. 27 is repeatedly and adaptively invoked according to the variation in the object (e.g. surface) S along the path of the profile curve provided thereon. Accordingly, in step 1204 of Fig. 28A, a sequence of one or more markers M_i , $i=1,2,\dots,N$, $N \geq 1$, is assigned to the variable MARKER_SET, wherein these markers are on the surface, S, and the markers are ordered according to their desired occurrences along a profile curve to be generated. Note that in one typical embodiment, the markers are generally provided (e.g. constructed and/or selected) by a user of the present invention. Moreover, for the present discussion, it is assumed that the tangents to the surface S corresponding to the markers M_i are tangents to S entered by the user. However, it is within the scope of the present invention that such tangent vectors may be provided automatically by, for example, determining a tangent of the direction of a parameterization of the surface S.

In step 1208 of Fig. 28A, the first marker in the set, MARKER_SET, is assigned to the variable, MARKER1. Subsequently, in step 1212, a determination is made as to whether there is an additional marker in MARKER_SET. If so, then in step 1216, the variable, INTRVL, is assigned a parametric increment value for incrementally selecting points on the profile curve(s) to be subsequently generated hereinbelow. In one embodiment, INTRVL may be assigned a value in the range greater than or equal to approximately 10^{-3} to 10^{-6} .

In step 1220, the variable, MARKER2, is assigned the value of the next marker in MARKER_SET. Subsequently, in step 1224, the curve interpolation program of Fig. 26 is invoked with MARKER1 and MARKER2 (and their corresponding user-identified tangent vectors) for thereby obtaining an interpolating curve, $C_j(u)$ between the two markers (where $j=1,2,\dots$, depending on the number of times this step is performed). Then in step 1228, an isocline boundary approximation is determined according to Fig. 27 using the values of MARKER1, MARKER2 and the interpolating curve, $C(u)$, for obtaining the isocline boundary approximation curve, $R_j(u)$.

Subsequently, in step 1240, the variable, u_VAL , is assigned the initial default value INTRVL for selecting points on the curves, $C_j(u)$ and $R_j(u)$. Following this, in step 1244, the variable INCRMT_PT is assigned the point corresponding to $C_j(u_VAL)$. Subsequently, in step 1245, the variable, S_PT , is assigned a point on S that is "closest" to the point $C_j(u_VAL)$. More precisely, assuming S does not fold back upon itself closer than $\epsilon > 0$, for some ϵ , a point on S is selected that is in a neighborhood less than ϵ of C_j . Note that since $C_j(u_VAL)$ may not be on S , by setting the value of INTRVL so that this variable's value corresponds to a maximum length along the interpolating curve C_j of no more than one-half of any surface S undulation traversed by this curve, then it is believed that the interpolating curve C_j will effectively follow or be coincident with the surface S . Subsequently, in step 1246, a determination is made as to whether the point INCRMT_PT is within a predetermined distance of S_PT (e.g., the predetermined distance may be in the range of 10^{-3} to 10^{-6}). In particular, the predetermined distance may be user set and/or defaulted to a system value that is changeable depending upon the application to which the present invention is applied. Accordingly, assuming that INCRMT_PT and S_PT are within the predetermined distance, then step 1248 is encountered wherein the point $R_j(u_VAL)$ on the isocline boundary approximation is determined and assigned to the variable, RIBBON_PT. Subsequently, in step 1252, an approximation to an isocline picket at $C_j(u_VAL)$ is determined and assigned to the variable, PICKET.

In step 1254, the tangent to the surface (more generally, object) S at the point $C_j(u_VAL)$ is determined and assigned to the variable, INCRMT_TAN, this tangent being in the direction of the parameterization of S .

In step I256, a determination is made as to whether the vectors, INCRMT_TAN and PICKET are sufficiently close to one another (e.g., within one screen pixel. If so, then a subsequent new point on the interpolating curve C_j is determined by incrementing the value of u_VAL in step I264. Following this, in step I268, a determination is made as to whether the end of the interpolating curve, $C_j(u)$, has been reached or passed. Note that the assumption here is that $0 \leq u \leq 1$. Accordingly, if u_VAL is less than 1, then step I244 is again encountered, and some or all of the steps through I256 are performed in determining whether the isocline ribbon point approximation, $R_j(u_VAL)$, is close enough to the actual ribbon point as theoretically defined in the Definitions Section hereinabove.

Referring again to step I246, note that if INCRMT_PT is not close enough to S , then an interpolating curve more finely identified with the actual points of S is determined. That is, the point, S_PT , is made into a marker and inserted into MARKER_SET, thereby causing new interpolating ribbon curves, $C_j(u)$ and $R_j(u)$ to be generated that will deviate less from S (assuming S is continuously differentiable). That is, step I272 is performed wherein a marker is generated for the point, S_PT , and this new marker is inserted into MARKER_SET between the current marker values for MARKER1 and MARKER2. Subsequently, the marker currently denoted by MARKER2 is flagged as unused (step I276), and in step I280, the most recently constructed interpolating curve $C_j(u)$ and any associated ribbon boundary curve $R_j(u)$ are deleted. Then, step I220 and subsequent steps are again performed for determining new interpolating and ribbon boundary curves, $C_j(u)$ and $R_j(u)$.

Note that steps I272 through I280 and step I220 are also performed if in step I256, INCRMT_TAN and PICKET are not determined to be sufficiently close to one another in the object space of S .

Referring again to step I268, if the end of the interpolating curve, $C_j(u)$, has been reached or passed, then it is assumed that $C_j(u)$ is sufficiently close approximation to points on S (between MARKER1 and MARKER2), and $R_j(u)$ is insufficiently close to an isocline ribbon for these points on S . Thus, if there are additional markers wherein an interpolating curve $C_j(u)$ and corresponding ribbon approximation $R_j(u)$ has not been determined, then the next pair of consecutive markers (of the marker ordering) in MARKER_SET is determined and various of the steps I220 and beyond are performed. That is, in step I284, MARKER1

is assigned the value of MARKER2, and in 1288, a determination is made as to whether there is a next unused marker in MARKER_SET. If so, then variations of the steps 1220 and beyond are performed as described above. Alternatively, if all markers have been designated as used, then in step 1292 the resulting curves $C_j(u)$, $R_j(u)$, for each used $j=1,2,\dots$, are graphically displayed and stored for subsequent retrieval. Note that the profile curves $C_j(u)$ may be optionally reparameterized so that these curves
5 may be parameterized collectively as a single curve, $\tilde{C}(u)$, with $\tilde{C}(0) = C_1(0)$ and $\tilde{C}(1) = C_N(1)$.

Figs. 29 and 30 provide high-level descriptions of flowcharts for modifying one or more surfaces (more generally geometric objects) by modifying isocline handles, ribbon tangents, and their associated isocline ribbons. In particular, for simplicity, the flowcharts of these figures assume that there is a composite surface, S_0 , that is provided (at least in part) by one or more subsurfaces, S_i , $i=1,2,\dots,N$, $N \geq 1$, where these subsurfaces S_i are connected to one another (e.g., patched together) along
10 common boundaries so that S_0 does not have disconnected portions. Accordingly, given such a composite surface, S_0 , the flowcharts of Figs. 29 and 30 can be described at a high level as follows. In Fig. 29, an isocline handle and/or a ribbon tangent having at least one geometric characteristic (e.g. length, direction, curvature, etc.) to be changed is determined along with the subsurfaces S_i that are to be modified to reflect the isocline handle and/or ribbon tangent changes. Subsequently, in the flowchart of Fig. 30, the modifications to the subsurfaces are computed and displayed in real-time as the user enters the modifications to
15 the selected isocline handle and/or ribbon tangent. Note that the computing of surface (more generally geometric object) modifications in real-time has not heretofore been feasible for surfaces in higher dimensional geometric objects in that the computational overhead has been too great. Accordingly, the present invention has reduced this overhead by providing a novel technique of computing blended surfaces which is very efficient and which generates surfaces that are fair.

The following is a more detailed description of Figs. 29A and 29B. In step 1400, if there are not profiles and isocline
20 ribbons corresponding to the entire boundary of each subsurface S_i , then make profiles and isocline ribbons that approximate the entire boundary of each subsurface S_i . Note that this may be performed using the program of the flowchart of Figs. 28. In step 1404, the isocline handles and ribbon tangents corresponding to markers on the surface S_0 are graphically displayed to the user.

In step 1408, a determination is made as to whether the user has requested to add one or more additional isocline ribbons to the surface S_0 , or extend an existing isocline ribbon having its profile curve on S_0 . If the user has made such a request, then step 1412 is performed to assure that in addition to any other markers added by the user, markers are added: (a) whenever a profile contacts a boundary of a subsurface S_i , and (b) so that profile curves will be extended in a manner that terminates each one on a boundary of a subsurface S_i . Moreover, additional markers may be also added at intersections of curve profiles. Thus, for these latter markers, there may be two distinct ribbon tangents associated therewith (i.e., one for each subsurface).

Subsequently, in step 1416, the program of Fig. 28 is invoked with each S_i , $i = 1, 2, \dots, N$ for thereby obtaining the desired additional profiles and isocline boundaries. As an aside, note that Fig. 28 need only be invoked with the subsurfaces S_i to which new markers are added.

10 In step 1420 following step 1416, all newly added isocline handles and ribbon tangents are displayed. Note that in some embodiments, only the isocline handles are displayed initially, and the user is able to selectively display the ribbon tangents as desired.

Subsequently, in step 1424, a determination is made as to whether the user has requested to add one or more additional markers within existing profiles. If so, then the additional new markers are added and at least the corresponding new isocline handles are determined for these new markers. As an aside, note that in one embodiment of the present invention, when a new marker is added to an existing profile, the profile will change somewhat since it is now exactly identical to the surface S_0 at another point and the interpolating curve generated (via Fig. 26) between consecutive markers of a profile is now generated using the newly added marker. Accordingly, a profile with one or more additional markers should, in general, conform more closely to the shape of the adjacent portions of the surface S_0 .

20 Subsequently, in step 1432, the additional new markers and optionally, their corresponding isocline handles and ribbon tangents, are graphically displayed to the user.

Note that it is not necessary for steps I408 through I420, and steps I424 through I432 to be performed sequentially. One skilled in the art of computer user interface design will understand that with event driven user interfaces, the processing of each new marker can be performed individually and displayed prior to obtaining a next new marker location from the user. Thus, consecutive executions of the steps I408 through I420 may be interleaved with one or more executions of the steps I424 through I432.

In step I436, a determination is made as to whether an isocline handle and/or a ribbon tangent is selected by the user for modification. Note that the identifier, ISO, will be used to denote the isocline handle and/or the ribbon tangent to be modified.

In step I440, the marker corresponding to ISO is determined and access thereto is provided via the variable, MRKR. Subsequently, in step I444, the collection of one or more subsurfaces S_1, \dots, S_n adjacent to MRKR are determined and access to these adjacent subsurfaces is provided by the variable, ADJ_SURFACES.

In steps I448 through I460, boundary representations of portions of the subsurfaces, S_i , adjacent MRKR are determined (step I452) and inserted into a collection of surface boundary representations denoted MOD_SET (step I456). In particular, for each of the subsurfaces in ADJ_SURFACES, a data representation of the boundary of the smallest portion of the subsurface that is adjacent to MRKR that is bounded by isocline ribbons is entered into the set, MOD_SET.

Finally, in step I464, the program of Fig. 30 is invoked for modifying, in real-time as the user modifies ISO, the portion of S_0 within the boundary representations contained in MOD_SET. In particular, the program of Fig. 30 is invoked with the values, MRKR and MOD_SET.

In the flowchart of Fig. 30, the high-level steps are shown for modifying in real-time the surface portions identified by the surface boundary representations in MOD_SET, wherein these surface portions are adjacent to the marker, MRKR. Accordingly, in step I504, the first (next) modified version of the isocline handle and/or ribbon tangent corresponding to the marker, MRKR, is obtained and assigned to, ISO. Subsequently, in step I508, all isocline ribbons containing the modified isocline handle and/or ribbon tangent of ISO are regenerated to reflect the most recent modification requested by the user. Note that this

is performed using the one-dimensional version of Formula (1), and modifying each such isocline ribbon along its extent between MRKR and the adjacent markers on each isocline ribbon containing MRKR.

Subsequently, in step 1512, the first (next) boundary representation in MOD_SET is assigned to the variable, B. Then in step 1516, the set of isocline ribbons for the (profile) boundary segments contained in B are assigned to the variable, \mathcal{R} . Note that \mathcal{R} includes at least one isocline ribbon containing the marker, MRKR.

In step 1520, a blended surface is generated that is delimited by the profiles of the isocline ribbons of \mathcal{R} . The formula used in this step is similar to Formula (4). However, there are additional functions, $Q_i(p)$, provided in the present formula. Note that, in general, the portion of a parameter space used in generating the surface, $S(p)$, of this step may have two, three, four, five or more sides (profile pre-images) that also have isocline ribbon pre-images. Thus, for the translation functions, $Q_i(p)$, for each isocline ribbon R_i of \mathcal{R} (wherein for the points p in the parameter space that are in the interior, I , to the pre-images of the profiles, P_i , for the isoclines R_i of \mathcal{R}) it is desirable that these points p be translated into points in the parameter space for R_i so that a corresponding point in the object space of the isocline ribbon R_i can be determined and used in the blending function of the present step. Note that the translation functions, $Q_i(p)$, preferably satisfy at least the following constraints:

- (a) $Q_i(p)$ is a continuous function for continuous surfaces;
- (b)
$$\lim_{p \rightarrow Q_i^{-1}(u, 0)} (Q_i(p)) = (u, 0)$$

That is, when a sequence of points in I converges to the pre-image of the profile point $P_i(u)$, then $Q_i(p)$ converges to the isocline ribbon parameter space point $(u, 0)$.

Subsequently, in step 1524, the surface $S(p)$ is displayed, and in step 1528 a determination is made as to whether there is an additional boundary representation in MOD_SET for generating an additional blended surface $S(p)$. If so, then step 1512 is again performed. Alternatively, if there are no further boundary representations, then in step 1532, a determination is made as to whether there is an additional user modification of the isocline handle and/or ribbon tangent corresponding with MRKR.

If there is, then at least the steps 1504 through 1528 are again performed. Note that the steps of Fig. 30 can be sufficiently efficiently performed so that incremental real-time changes in the isocline handle and/or ribbon tangent for MRKR designated by the user can be displayed as the user continuously modifies this isocline handle and/or ribbon tangent.

4. A Geometric Design User Interface

5 The general principles described above form a basis for a novel user interface for computer aided geometric design.

In one user interface embodiment for the present invention, a user interface may be provided for defining isoclines. Using such an interface, a designer may, for example, require that an isocline be perpendicular to a given light direction along an entire profile curve so as to create a reflection line, as one skilled in the art will understand. More generally, the novel user interface may allow for various constraints to be input for generating isocline ribbons, isocline handles and/or ribbon tangents that satisfy such constraints. In particular, the user interface allows for global constraints such as light direction, curvature, tangency, level contours, dihedral angle functions with a plane, etc., as one skilled in the art will appreciate.

15 In one embodiment of the user interface, the user will start with a given geometric object, for example a cylinder. The user may then inscribe a profile curve on the cylinder by placing markers at various points on the cylinder. The profile tangents and/or isocline handles may be defaulted by adopting the slope information from the cylinder. For example, at each marker, the profile tangents are in the plane tangent to the cylinder at the marker.

The user may then select and modify the markers, add additional markers, and/or modify the position and the direction of the isocline handles and/or ribbon tangents. As the isocline ribbon is accordingly modified, the cylinder (more generally, geometric object) will reflect the changes in the modification of the isocline ribbon. Additional profiles and markers may be added in this manner until the desired shape of the geometric object (derived from the cylinder) is obtained. An example of these steps is illustrated in the flowchart of Fig. 31. That is, the user selects a graphically displayed surface (more generally, geometric object) in step 1904. Subsequently, in step 1908, the user constructs a profile curve on the selected surface (object).

Subsequently, in step 1912, an isocline ribbon (or at least the isocline boundary) is generated for the profile. Note that this ribbon/boundary can, if desired, be generated substantially without additional user input. That is, an isocline ribbon/boundary may be generated from the tangency characteristics of the surface upon which the profile resides. In particular, for a parametric surface (more generally geometric object), the parametric tangents on the surface at points on the profile can be utilized to generate an isocline ribbon/boundary for the profile. Thus, surface neighborhoods on one side of the profile curve may be used to determine a first isocline ribbon/boundary for a first surface having the profile, and if the profile is on the seam between the first surface and a second surface, then surface neighborhoods on the other side of the profile may be used to determine a second isocline ribbon/boundary.

Additionally, note that other surface characteristics may be preserved in an isocline ribbon/boundary. For example, in addition to preserving the parametric tangents at profile curve points, the isocline ribbon/boundary may also optionally preserve the surface characteristics such as curvature, high order (≥ 2) derivative continuity with the surface. Note, however, it is within the scope of the present invention that further surface characteristics can be preserved in the isocline ribbon/boundary.

In step 1916, the generated isocline ribbon/boundary may be used to modify the surface(s) having the profile curve as discussed hereinabove with reference to the programs of the flowcharts of Figs. 29 and 30.

In some embodiments of the user interface, an operation is provided to the designer wherein a common boundary between two object space surfaces can be selected and the operation automatically forces the surfaces to join at a higher order continuity criteria (e.g., curvature continuity) than that of tangent plane continuity. For example, a higher order continuity constraint imposed on an isocline ribbon deriving from one of the surfaces at the common boundary, can be used to similarly constrain an isocline ribbon for the other surface having the common boundary. Accordingly, this operation helps alleviate the so-called "Mach band" effect in which the human eye detects discontinuities in curvature in some instances.

Other user interface operations provided by the present invention are:

(a) *Rounderizing*, which is a tweaking operation that modifies an existing surface to round off pointed edges, or to create darts (i.e., surfaces that are smooth except at a single point, where the surface kinks) that dissipate sharp edges. Such operations can be performed using the present invention by positioning profile curves on the surface on opposite sides of a sharp edge and blending smoothly between the profiles (e.g., using Formula (1) as described in Section 2.3.5; and subsequently eliminating the surface in between the profile including the sharp edge.

(b) *Embedding*, which is an iterative user interface procedure that can take one finished model, scale it, and perhaps rotate or otherwise deform it to fit into part of another model.

4.1. Defining the Isocline via Markers, Profiles and the User Interface

Explicit profiles are the profile curves that express a designer's intent. Explicit profiles may be unconstrained (freeform) or partially constrained (trim). Implicit profiles may be visible boundaries between surface patches caused, for example, by a surface discontinuity (i.e., a kink or curve defined between an end surface of a cylinder and the cylindrical side thereof).

Implicit profiles are created automatically when the user introduces, e.g., a surface discontinuity. All profiles in a model are either explicit or implicit.

4.1.1 Creating Markers

Profile markers and handles are created in the following ways:

- A. Markers are automatically created at the ends of new implicit and explicit profiles.
- B. Inserted by a designer (e.g., by double clicking at a point) on an explicit profile. To the designer, he/she is inserting a point on the profile. The newly placed marker only minimally or not at all changes the shape of the profile in the profile segment containing the new marker. Subsequently, profile and isocline handles are determined according to the shape of the profile and surface(s) attached at the new marker.

A marker may be identified with a plurality of coincident points on the same profile (e.g. a profile that loops back and attaches to itself). Such an identification of the marker with the plurality of profile points cannot be broken, except by deletion of the marker. In the case of two or more profiles meeting at a common point having a profile marker, such profiles each have a marker at the common point and the markers are constrained to maintain coincidence so that moving one marker will move both.

Profile markers inserted by the designer may be inserted for providing profile handle points, or for setting specific isocline values. Note that a profile handle point may have a set of constraints on its isocline handles; i.e., isocline handle may inherit value(s) by an interpolation of the nearest two adjacent isocline handles.

4.1.2. Viewing Markers and Profiles

Profile and isocline handles may have various constraints placed upon them, wherein these handles may be displayed differently according to the constraints placed upon them. In particular, the following constraints may be placed upon these handles:

- (a) constrain a handle to a particular range of directions;
- (b) constrain a handle to a particular range of magnitudes;
- (c) constrain a handle to lie in a plane with other handles;
- (d) constrain a handle to a particular range of curvatures;
- (e) constrain a handle with a transform of another handle, e.g., identical rotations and/or translations.

The designer can choose to display the constraints through a display request for the properties of geometric objects. In one embodiment, different colors may be displayed for the different types of constrained profile markers. For example, handles having no variability (also denoted herein as "fully constrained") may be displayed in blue.

In some embodiments of the user interface, vectors are "grayed out" that are constrained to thereby demonstrate to the designer that such vectors cannot be changed. For example, in one embodiment, fully constrained handles are typically grayed out.

4.1.3 Connecting Profiles Together

In one embodiment of the user interface, it supports the linking together of two or more profiles that intersect at the same X,Y,Z location. Such intersection points are denoted "tie points," when the parameterization of the point on each profile is invariant under profile modifications. Note that such tie points may or may not have markers associated therewith. When such a tie point is modified, all corresponding profile curve points associated therewith at the tie point are modified as a group. Such a tie point may be an endpoint of a profile or an internal (i.e. "knot") point.

Alternatively, a profile marker of a first profile may be constrained to lie within the object space range of a second profile (either implicitly or explicitly). For example, referring to Figs. 42A and 42B, the user interface may provide the user with the capability to slide a profile marker 2002 (contained on a first profile 2003 and a second profile 2004) along the second profile 2004 for thereby changing the profile 2004 of Fig. 42A into the profile 2004 of Fig. 42B when the marker 2002 is slid along the profile in the direction of direction arrow 2006. Such a slidable marker 2002 is denoted as a "slide point."

Profile intersections are either "slide" or "tie" points. Moreover, these different types of points may be distinguished graphically from one another by different colors and/or shapes. Note, if a profile slides along another profile, and the isocline ribbon for the sliding profile is used to compute a blended surface, S , then S will be recomputed.

4.1.4 Creating Markers and Profiles

The user interface may support the creation of a profile curve in a number of ways:

A. Sketch the profile on the surface similarly to the data driven technique of Figs. 28, wherein additional markers may be provided for tying the profile to the surface within a predetermined tolerance. Alternatively, in a second embodiment, a profile can be sketched across one or more surfaces by having the user select all markers for the profile. Note that in either case, a profile may be sketched across one or more surfaces. Moreover, in the second embodiment, the user interface supports the following steps to successfully create a profile.

(A1) Marker points are specified which lie on the surface(s). A fitted curve is generated through the points in parameter space and then this fitted curve is evaluated to obtain a corresponding image curve in object space.

(A2) For each surface that the new profile crosses, the user may specify the profile type (freeform or trim) for the portion of the profile crossing the surface. A profile sketched on a surface either trims one side of that surface or splits that surface into two new surfaces. Accordingly, note that any (non-degenerate) profile that enters the interior of a surface must cross the surface's boundary at an entering point and an exiting point. That is, the two surfaces along a common profile boundary are always linked to the profile, wherein, with respect to this profile, these surfaces may be characterized as follows: (a) one surface is a trim surface and one is a non-trimmed surface (also denoted a "freeform surface"), or (b) two freeform surfaces.

B. Copy a Profile: A designer selects a profile to copy. The profile is copied to a buffer (denoted a clipboard). The user then selects the mode of copy (examples: keep profile handles, or adapt profile handles to the geometry where the profile is to be copied). The user selects where to position the new profile (which may be additionally scaled, rotated, and/or mirrored, etc). The user selects a location for the new profile. Note that the new profile can be constrained by constraints on the original profile. For example, the new profile may be mirrored about a plane from an existing profile, such that any change to the original profile will result in a change to the copy.

When a new profile is created, profile markers are automatically generated at both ends of the new profile. Profile handles and isocline (ribbon tangent) handles are derived from the geometric characteristics of the surfaces that the new profile splits.

4.1.5 Modifying Markers and Profiles

Modification of markers and/or (profile, isocline) handles is dependent on the constraints placed upon the markers and/or handles. They may be constrained in one of the following ways:

A. Interactive modification by selecting a handle (profile or isocline) at a particular marker, m , and moving the handle endpoint so that it is constrained to lie in a normal plane, i.e., either a plane normal to a surface having a profile

containing m , or a plane defined by the isocline and profile handles of the profile at m . A pop-up property sheet is also available for the user to key-in specific numerical values for magnitude and angle for modifying a selected profile and/or isocline handle.

5 B. For markers that are constrained so that their pre-images lie within the pre-image of a profile, the marker will may slide along such a parent profile via, e.g., interactive modification by dragging such marker points. Note that for positional unconstrained markers, the marker points may be moved freely (i.e., under a user's direction and without other restrictions) along the parent profile(s) upon which such markers reside. Additionally, note that the user can select multiple profile marker points by clicking on each, or by selecting all markers within a designated region (e.g., bounding rectangle). Accordingly, the user is able to move a display pointing device (e.g., a mouse) which will then cause all of the selected markers to uniformly move in a corresponding direction to the movement of the display pointing device. However, movement of the markers depends on the constraints set on these markers. For example, a constrained marker will only move within the limits placed upon it. Thus, if a first selected marker moves only within a first profile and a second selected marker only moves within a different second profile oriented orthogonally to the first profile, then depending on the direction of movement desired, one of the following occurs:

- 15 (i) the first marker is able to move, but the second marker can not;
- (ii) the second marker is able to move, but the first marker can not;
- (iii) both the first and second markers are able to move;
- (iv) neither marker may be able to move.

 C. Marker and handle constraints may be set by default rather than explicitly by the user. A pop-up property display form allows the user to set or remove specific constraints.

20 D. Additional constraints on profile and/or isocline handles may be set that are dependent on the characteristics of other geometry. For example, profile and isocline handles can be constrained to be normal or parallel to a selected reference plane. Note that the position of a profile marker also can be constrained to be dependent on characteristics of other geometry.

For example, a marker can be constrained to lie on a parting plane. That is, a plane of front/back symmetry for designing a bottle. Another example of these constraints is in generating symmetric designs, i.e., a profile marker copy that is reflected about a parting plane will be constrained to be symmetric to the parent profile marker.

- 5 E. Surfaces adjacent to a profile may have to satisfy either C0, C1, or C2 continuity, wherein C0 is positional, C1 is tangency, and C2 forces smooth surface blends. One constraint that can be set on a marker is to force C1 continuity between surfaces surrounding the marker by maintaining equal length tangent vectors interior markers.

Freezing (eliminating the ability to modify) the profile and isocline handles at a marker will cause the profile segment containing the marker to rebuild based on the profile handles of the next two closest markers, effectively changing this marker's handles to reflect the curve built by the two markers on either side.

- 10 For the most part, modifying profiles is a function of user interface techniques for modifying the profile marker positions and handles that control the shape of the profile. The following are examples of such user interface techniques.

A. Direct method: Profiles are modified directly on an object space (3D) model. This is done by modifying the profile markers and handles that make up the profile.

- 15 If the designer modifies a trim profile, the profile always lies within the parameter space of the surface it is trimming. That is, the trim profile needs to be modified in the context of its original, overbuilt surface within which it is embedded. Selecting a trim profile (or one of its components) to modify causes the overbuilt, construction geometry to be highlighted. It remains highlighted while the user is modifying the trim profile.

- 20 A designer may have the option to turn on profiles and modify them using the direct method. For example, modifying a profile that defines an overbuilt surface will cause the overbuilt surface to be updated. Since the profile that trims this overbuilt surface is constrained to lie within the parameter space of the surface, the trim profile is also recomputed.

B. Design Ribbon method: This method is used to modify a specified region of a profile. It allows, for example, the user to simplify the designer's interactions by modifying a profile in one view that is complex in another view. The designer

identifies two markers that lie on the same profile. The profile segment(s) between the two markers are extruded in at least one graphical view of the profile, thereby creating a design ribbon (not to be confused with an isocline ribbon). The design ribbon is a simple extruded surface (i.e., a curve which is swept along given directions to generate a surface; for example, for markers at the ends of a profile, offset the corresponding profile handles by their corresponding isocline handles to obtain boundary handles and interpolate an isocline boundary, e.g., by a lower dimensional version of Formula (2); the surface having a perimeter consisting of the profile, the isocline handles, and the isocline boundary defines the new extruded surface). The three-dimensional profile segments identified between the markers always lie within the pre-image of this design ribbon. The user modifies the profile in one of two ways:

(B1) Modify the two-dimensional driving curve from which the design ribbon was extruded, and which is instanced at the end of the ribbon, wherein, by default, this curve is a two-dimensional representation of the three-dimensional profile segments defining the design ribbon. The user may "simplify" the driving curve by selecting a subset of the two-dimensional points. Each time the user modifies a driving curve point, the ribbon is updated and the three-dimensional profile is modified to lie within the parameter space of the modified ribbon. Operations on the driving curve include any of those listed in the Profile Marker section (point/slope modification, insert, delete, etc.).

(B2) Modify the two-dimensional profile points within the design ribbon. The user directly modifies the two-dimensional profile in a view that is perpendicular to the primary view. The two-dimensional points always lie in the parameter space of the ribbon. Operations include any of those listed in the Profile Marker section (point/slope modification, insert, delete, etc.).

Only one design ribbon may exist per surface per profile segment. Design ribbons may be created, modified, and deleted. Once they are created, they are persistent, i.e. they remain unmodified until a designer modifies the same segment at a later point in time. A design ribbon is displayed only when a designer is modifying it. A single profile may have multiple ribbons corresponding to multiple surfaces containing the profile.

Note that modifying a profile using the direct method deletes any design ribbons spanning the points being modified.

This invalidates the ribbon and requires a designer to re-specify the ribbon.

C. Move Profile(s): The designer selects and moves two or more profiles in unison. That is, this user interface command selects all of the profile markers on a profile (or segment thereof) and moves them together as a unit.

5 D. Merge Profiles: The designer may sketch a new profile and attach it to an existing profile so that an endpoint of each profile is coincident. Additionally, the designer may specify which segment or segments of the existing profile to delete. Subsequently, the new profile and the remaining connected portion of the existing profile having the coincident end point with the new profile are merged. Note that merging profiles causes the set of the respective profile handles, isocline handles and ribbon tangents for each of the two coincident endpoints to be combined into a single
10 such set.

E. Split profile: Split one profile into two at a single point, p . An endpoint of each of the two new profiles is constrained to be coincident at p .

4.1.6 Deleting Markers and Profiles

Deleting a profile marker is always possible, except at the endpoints of a profile. However, in some embodiments of
15 the present invention may need to replace a marker with a marker having constraints if it is needed for maintaining smooth patches. Note that a new constrained replacement marker may or may not be in the same location as the previous marker.

If an entire profile is deleted, then the user interface both highlights any dependent geometric object(s) and requests user confirmation before deleting the profile and the dependent geometric object(s). Accordingly, note that the present invention retains sufficient dependency information regarding dependencies between geometric objects in a model so that for modifications
20 of an object that is used for deriving other objects, appropriate additional modifications can be performed on these other objects automatically.

4.1.7 Profile Markers and Handles

Note that there are typically two profile handles, two isocline handles and two ribbon tangents for a profile marker, i.e., a profile handle, an isocline handle and ribbon tangent per surface on each surface having the profile as a boundary curve. However, there may be more handles associated with a profile where several profiles converge or fewer if the profile is the edge of a surface.

4.2 Isoclines and the User Interface

The slope of an isocline handle controls surface tangency at a marker and at a surrounding portion of the profile containing the marker. The magnitude of an isocline handle controls the fullness of the dependent surface. That is, how much the surface bellies out. An isocline handle may be constrained to be offset from another isocline handle (i.e. -10 degrees from other side). An isocline handle can be calculated at any point along a profile (by inserting a marker on the profile).

4.2.1 Creating an Isocline Handle

The user interface supports the constraining of isocline handles relative to one another. Such handles can be forced to always be tangent, of equal magnitude, or offset by some amount. In one embodiment of the present invention, the user interface provides a pop-up menu to display and change isocline handle constraint values, such as length and direction.

4.2.2 Modifying an Isocline Handle

If the user slides a profile marker along a profile, the user may fix the isocline handle for the marker, thereby causing the surfaces adjacent to (and dependent upon) the profile to change or have the isocline handles interpolated between the nearest two isocline handles on the profile (which case implies that the dependent surfaces are not affected)

4.2.3 Deleting an Isocline Handle

The user interface supports the deletion of isocline handles. The following steps apply when deleting an isocline.

4.3 Special Geometric Objects and the User Interface

The present invention provides for the creation and manipulation of a number of specialized geometric object types that can substantially facilitate the design of objects such as containers.

4.3.1 Label Surfaces

A label surface is a special case of a trimmed surface. The special case aspects of a label surface are:

- 5 (i) there is a "watershed" profile that runs from the bottom to the top of the label;
- (ii) there are label curves between which the corresponding label surface is ruled (e.g., label curves 132 of Fig. 11);
- (iii) there is a boundary (trim) profile (e.g., trim profile 138 of Fig. 11).

The key difference that makes a label surface different from other trimmed surfaces is that the original surface (from
10 which the label surface is trimmed) is a ruled surface. In particular, the label surface defining curves are constrained such that a ruled surface is maintained within the boundary of these defining curves.

Note that other surfaces may be blended to a trim profile for a label, but the trim profile can only be modified in a manner that insures that it bounds a ruled surface.

In one embodiment of the present invention, a two dimensional "rolled out" representation of the label surface can
15 be generated. That is, the surface can be associated in a one-to-one fashion with the plane by rolling it flat. Such a representation simulates a label surface in which a designer can thereon create a piece of artwork that can subsequently be wrapped on a container.

4.3.1.1 Creation of a Label Surface

To create a label surface, an overbuilt surface to be trimmed must be a ruled, approximately developable surface, i.e., a ruled surface in which all surface normals on any ruling are parallel. Subsequently, the user then follows the normal trim
20 surface steps; i.e., sketching a profile on the ruled surface, generating the (sub)surfaces on both sides of the profile (i.e., the label surface and the portion of the surface to be trimmed away), trimming the label surface and blending other surfaces to the trim profile.

Note that the trimmed away surface portion is hidden from normal viewing (i.e., it is no longer a part of the visible model).

The following procedure may be provided for generating a label surface. At a high level, the steps for this procedure are:

- 5 (i) Make sure the surface is ruled. That is, the user interface supports automatic modification of user selected profiles so that these profiles satisfy 4.3.1(i) and (ii). In particular, to perform this step, the following substeps are performed:
 - (ii) The user sketches a boundary profile on the surface defining the bounds of the label;
 - (iii) Construct a graphical representation of a label (i.e., a ruled surface having text, artwork, and/or designs thereon);
 - 10 (iv) Allow the user to graphically apply the label representation to the label surface (or a representation thereof). In particular, the user interface for applying the label representation may automatically attach the label representation to the label via a grouping type of operation so that the label representation maintains its position on the label surface during, e.g., label surface rotations, translations, scaling operations, etc.;
 - 15 (v) Allow the user to undo the design when the label surface and/or the label is not satisfactory.

4.3.1.2. Modification of a Label Surface

Modification of the label surface components is somewhat different than those of a trimmed surface.

The portion of the watershed profile that is a straight line segment is constrained to remain straight.

The boundary opposite of the watershed (the "other side, of the parent ruled surface) cannot be modified. It is simply
20 a straight line segment between the top and bottom boundaries. The top and bottom boundary profiles can be modified.
They are constrained so no additional free profile markers can be inserted on them. Also, the profile marker at the end away from

the watershed is constrained to move only to maintain a ruled surface. It can be extended (extrapolated along the same curvature) and the angles at its endpoints can be adjusted - again, as long as it maintains a ruled surface.

4.3.1.3. Deleting of a Label Surface

Deleting a label surface removes the constraints on all of the profiles used in creating the label surface. Additionally,
5 all of the construction geometry for the label surface that is invisible to the user will also be deleted. The constraints for maintaining a ruled surface will also be removed. Thus, the remaining geometric objects are then freed from the label surface constraints, and can be modified in ways not previously available.

4.4. Hole Tool User Interface

The present invention may also provide a user with a novel computational method that helps the user add a hole to
10 a geometric model (such as for adding a handle to grasp to a non-handled bottle). The information required to add a hole to a model using this procedure includes: a loop of profile segments on a front surface, a loop of profile segments on an opposite back surface, the type of each profile in a loop of profiles (freeform or trim), and optional profile(s) to shape the interior of the hole.

4.4.1. Creation of a Hole

15 The hole creation tool guides the user through a series of steps to add a hole. Figs. 22A-22C illustrate the procedure for creating a hole 600 (Fig. 22C) on a geometric object 604 using the present invention. The corresponding steps performed for creating the hole 600 are as follows:

- (a) Sketch a profile loop 608 on the front surface 612;
- (b) Sketch a profile loop 616 on the back surface (optionally project the profile 608 to the back surface).

20 Note that isocline handles are automatically placed on both profiles 608 and 616.

- (c) If one or more of the profiles for one of the profile loops 608 and 616 are freeform profiles, then the user may add new profiles (to complete such a profile loop), and/or profiles merge corresponding to such a loop

whereby these profiles are constrained so that they are utilized as if they were a single profile. Accordingly, once the profile loops are constructed, then surfaces may be skinned between the loops to thereby replace the original surface occupying the hole.

If a trim profile loop is specified, the surface region inside the profile loop is trimmed.

- 5 If specified, the hole creation procedure uses additional profiles to place and shape surfaces on interior boundaries for the hole. Otherwise, surfaces are skinned automatically between the front and back profile loops.

4.4.2. Modification of a Hole

Modifying a hole is a function of modifying profile markers and handles that make up the geometry of the hole.

4.4.3. Deleting of a Hole

- 10 Deleting a hole is also a function of deleting the components that make up the geometry of the hole; i.e., profiles and other geometry for the hole.

4.5. Smoothness Considerations via the User Interface

We consider the order of transition between adjacent surfaces (which meet at the profiles). This section does, however, contain some broader implications for the general theory in Section 2.

15 4.5.1. Continuous Profiles via the User Interface

Given two profile curves that intersect, derivative continuity across an intersection point may be assured if several conditions are met:

- (a) an end point of one profile is coincident with an end point of the other profile (positional continuity);
- (b) the blending functions B_i used in generating the profiles (as per Figs. 26 and 27) are equal at the
- 20 intersection marker; and
- (c) the profile handles at the intersection marker are collinear and equal length.

Tangent directional continuity is a weaker condition that can be satisfied if condition (c) above is changed to:

(c*) the profile handles at the intersection marker are only colinear. The magnitudes of the profile handles may differ in this case.

The designer may intentionally produce a kink at a marker by *breaking* the collinearity of the two profile handles at the intersection marker. This means that the two profile handles do not have a common direction.

5 **4.5.2. Continuous Surfaces**

The notion of tangent plane continuity between surfaces may be defined as follows: for each point p of a boundary between two surfaces S_1 and S_2 , the tangent plane, $T_1(p)$, of S_1 at p is identical to the tangent plane, $T_2(p)$, of S_2 at p . To achieve tangent continuity between surfaces across a profile boundary therebetween, it is necessary that the isocline handles (for each of the surfaces) at each marker on the profile boundary lie in a common plane with each other and the profile handle at that marker. If this is not done then a kink in the surface along the profile will be created.

Note that when there are two or more surface patches to be generated wherein these patches must be constrained to meet at a common marker point, p , the present invention may automatically generate isocline handles, denoted "common direction handles." That is, for each profile, P (having p) used in defining one or more of the surfaces, there may be a corresponding automatically generated common direction handle which is a vector, V , oriented from p , wherein V is perpendicular to the profile handle of the profile P , this profile lying in the common plane formed by the profile handles for the other profiles also having the point p . Further note, the user interface supports allowing the user to either display or not display the common direction handles.

Note that it is not necessary to have profile handles and isocline handles that match for profiles that adjoin at a common marker in order to achieve smooth surfaces there, only that they all lie in a common plane. In Fig. 18, there are three profile curves 404, 408 and 412 for the surfaces 416 and 418. Each of the three profile curves meets at the profile marker 420, and each of the profiles has a corresponding isocline ribbon 424 (for profile 404), 428 (for profile 408), and 432 (for profile 412). Additionally, the profile and isocline handles associated with the profiles 404, 408 and 412 and the marker 420 are:

- (i) profile handle 436 and isocline handle 440 for profile 404;
- (ii) profile handle 444 and isocline handle 448 for profile 408;
- (iii) profile handle 452 and isocline handle 456 for profile 412.

Thus, if the profile and isocline handles 436, 440, 444, 448, 452 and 456 all lie within the plane 460 (indicated by the dashed rectangular portion), then the surfaces 416 and 418 smoothly join at the marker 420.

At any marker, two isocline ribbons are likely to meet in the way that two profiles may meet, that is, two isocline ribbons may have a common isocline handle as an edge for each of the ribbons.

To achieve tangent plane continuity between different (blended) surface regions S_1 and S_2 (Fig. 43) joined by a composite profile, not only is tangent continuity across profiles P_1 and P_2 needed, but tangent continuity between adjacent ribbons R_1 and R_2 is also needed. That is, for the profiles P_1 (between markers 2010 and 2014) and P_2 (between markers 2014 and 2018), the respective ribbons R_1 and R_2 , when thought of as surfaces, must be tangent plane continuous, and share a common isocline handle 2022. Note that in most cases, tangent continuity between ribbons is equivalent to tangent continuity between profiles and tangent continuity between ribbon boundaries that is required for smooth transitions across surface patch boundaries. Moreover, the user interface of the present invention provides techniques for assuring tangent plane continuity between ribbon boundaries wherein these techniques are substantially identical to those used for assuring tangent plane continuity between profiles. Thus, the present invention can provide tangent plane continuity between adjacent surfaces generated from isocline ribbons according to the present invention.

In some circumstances, it is possible to break the continuity of composite ribbons intentionally, thereby causing a crease across the surface generated from the ribbons wherein the crease does not correspond with a coincident profile along the crease. However, in some embodiments of the present invention, an “*implicit profile*” can be created that is coincident with the crease.

4.5.3. Curvature Continuity

The visual quality of a surface depends not only on tangent plane continuity, but also on higher order derivatives. A user can be acutely sensitive to discontinuous changes in surface curvature, especially if the surface is rendered with specular highlights or reflected texture mappings, which is common in simulating realistic scenes. The user may perceive a distracting visual artifact known as a "Mach band." Accordingly, raising the order of continuity between transitions to that of curvature continuity ameliorates this.

Analysis has shown that the curvature of the surface defined by Formula (1) or Formula (4) depends on the second derivatives of the B_i and the S_i . The dependencies of the B_i are non-trivial and it is advantageous to choose the blending functions so that their second derivatives are zero and let the surface functions S_i determine the curvature. The cosine squared function of Section 2.1 fulfills this condition. There also exist certain quintic polynomials that are satisfactory. For example, the polynomial $B_i(x)$ of Formulas (3.1).

If the curvature of a blended surface generated from Formula (1) or Formula (4) depends only on the S_i (e.g., $B_i'' = 0$), it is then possible to raise the curvature order between the bounding surface patches S_i analogous to the methods in the previous section for achieving tangent continuity. To do this, simply define the corresponding profiles and isocline handles so they match in their second derivative at each marker along the profile boundary. Note, however, that each profile handle may be considered as a linear function of one parameter and therefore has a zero second derivative. Thus curvature continuity is achieved; albeit by making the curvature across the profile "flat," i.e., zero. This is useful at points where there is an inflection point on the profile, but can be undesirable elsewhere. To rectify this situation, the linear handles may be replaced with curved *ribs*, such as parabolic arcs. Accordingly, the handles now become arcs, and at the markers, the curvature is made to match that of the given arc.

By extending the concept of providing a nonzero curvature to all handles, e.g., profile, isocline and boundary handles, along with the zero second derivatives of the blending functions and the effects of the Mach banding can be mollified.

4.5.4. GI Continuity Using Roll, Yaw and Magnitude Representations.

The present invention also provides a user interface method to specify handle vectors (e.g., isocline handles) relative to a corresponding profile curve, wherein GI continuity (as defined in the Definitions hereinabove) between surfaces joined together by the profile is assured. This method, which is denoted herein as the roll-yaw method, specifies a vector V in terms of three scalar terms called roll, yaw and mag (magnitude), wherein roll and yaw are determined at a point P on a curve using the tangent vector T at the point P , and a vector N normal to the curve at the point P . The yaw component of the vector V represents the angular deviation from T at P . For instance, if the vector V is in a direction perpendicular to T , the yaw value (in at least one framework) is 0° , and if the vector V at P is in the same direction as T , then the yaw value is 90° . Regarding the roll component of the vector V , this scalar represents the amount of angular rotation about T as the axis of rotation, and wherein the baseline axis for measuring the angle is the vector N at P . Accordingly, the vector N represents 0° of roll and the rotational range extends from -180° to 180° using the right-hand rule, as one skilled in the art will understand. Regarding the magnitude component of vector V , this is simply the length of the vector V . Note that any vector expressed in terms of three-dimensional Cartesian coordinates can be transformed one-for-one into the roll, yaw, mag notation for a given T and N .

Note that the vector N may be selected from among vectors in the plane normal to T . However, this does not precisely define N . Thus, several methods may be used to define N . A first such method for defining N is simply to choose a constant vector VC and then determine N by the following equation: $N = T \times VC$. This method, however, produces an undefined value for N when T and VC are colinear. To provide values for N where this equation yields a zero vector, N can be approximated in a topological neighborhood of the colinearity. Alternatively, in a second method of generating N , the Frenet-Serrat frame of the underlying curve may be chosen, as one skilled in the art will understand. However, the Frenet-Serrat frame may be discontinuous at inflection points along the curve. Accordingly, the present invention provides a method for creating a minimally rotating reference frame for a complex (i.e., three-dimensional) curve that obviates difficulties in defining the vector N regardless of the orientation or shape of the curve and its tangent vector T .

As previously mentioned the roll-yaw method provides a novel way to achieve G1 continuity across a profile. As an example, consider the geometry illustrated in Fig. 44, wherein a profile P along with left and right isocline ribbons LR and RR are shown. Each of the isocline ribbons LR and RR has two corresponding isocline handles at its ends, i.e., HLI and HL2 for LR, and HRI and HR2 for RR. Assuming the profile P endpoints have handles denoted HPI and HP2, for any point pp on the profile, continuity across the profile for surfaces bounded thereby is determined by the interpolated isocline values IL and IR. Further, IR is interpolated (according to the techniques of the present invention) from HRI and HR2, and IL is interpolated from HLI and HL2. Thus, one skilled in the art will understand that for G1 continuity across the profile P, IL and IR must at least be in opposite (colinear) directions. Further, it can be shown by one skilled in the art that if IR and IL are formed using a cubic Hermite interpolation between HLI and HL2 for IL, and, HRI and HR2 for IR that the conditions for G1 continuity are that HLI and HRI must be equal and opposite vectors. Further, the same must be true for HL2 and HR2. However, if instead of interpolating the isocline values IL and IR in Cartesian space, the interpolation is performed in (roll, yaw, mag) space, G1 continuity can be achieved with a less strict condition, namely, that the roll value of HLI and HL2 must be the same. Accordingly, this is equivalent to saying that HLI, HRI and HPI must be no more than coplanar (with the same being true for HL2, HR2 and HP2) in order to guarantee G1 continuity everywhere on the profile P. Further, note that similar conditions may be imposed if the isocline handles are curved rather than straight. In particular, tangent vectors to the isocline handles at their common points with the profile P may be used in place of any corresponding isocline handle vector represented in Fig. 44. Thus, as one skilled in the art will appreciate, computational steps can be provided that embody the roll-yaw method for, if necessary, converting from Cartesian vectors to roll, yaw, mag vectors, and then assuring that the above described coplanar constraint is satisfied for guaranteeing that surfaces are G1 continuous across the profile P.

4.6. Embedding Models Within Models

The present invention allows parts of a surface bounded by profiles to be designed separately from one another. For example, a triangular portion of a surface may be designed as a free standing surface model. That is, a designer may add profiles and isocline ribbons as desired until a satisfactory design of the model is obtained (using barycentric mappings as one skilled in the art will understand). Afterwards this piece may be distorted, rotated and fit into a triangular portion of another model. Hence, a finely detailed model may be designed and embedded into another model. By maintaining links this process can be used for level of detail management. That is, for example, when the model is viewed from a distance, the detailed portion is unneeded for display, but as the viewer moves closer the embedded object is linked in for the extra detail it affords. Two examples of types of embeddings follow in the next subsections.

4.6.1. A Rounderizing Technique

Referring to Fig. 19, a small blended surface rounds an edge 482 between two intersecting surfaces 484 and 486. This blended surface 480 is blended from the thin surface strips 488 and 490 whose pre-images are a “small” offset from the pre-image of the edge 482 in parameter space. This process is a straightforward application of Formula (1) where the two surfaces 484 and 486 are blended using their common parameter space (not shown).

The new surface types lead to new evaluation routines that are especially efficient in special cases described.

5. Evaluation

We will first consider the evaluation of the two-edge blend, recognizing that other forms derive from this fundamental form. Because of its importance we will recall Formula (1), which is

$$S(u,v) = S_1(u,v) B_1(u,v) + S_2(u,v) B_1(u,v). \quad (1)$$

There are both blending functions B_i and isocline ribbons S_i to determine when evaluating the surface S . The blending function is calculated as a univariate function of distance in the parameter space. As discussed in Section 2, the evaluation of the distance

function varies considerably depending on how complex the pre-image is in parameter space. Once determined, the actual blending value can be calculated by a simple table look up; that is, the blending functions are tabulated to a sufficiently high resolution and stored in memory where they can be indexed by the input variable. Consider the function of $B_i(x)$ of Formula (3.1). Evaluate this function at $x = 0, .01, .02, \dots, 0.99, \text{ and } 1$. These 1001 values are stored as an array. When a point X is given, it is used to locate the nearest point in the array, e.g., between .52 and .53. Subsequently, $B(.52)$ or $B(.53)$ are used as the function value.

There are many techniques that may apply based on what the distance and isocline ribbon functions are. The present discussion is focused on a method that assumes a simple model computationally, but nevertheless, retains considerable design flexibility. The isocline ribbons 508 (S_1) and 516 (S_2) will be given as in Fig. 20. These are parameterized from 0 to 1 in both u and v parameters. For each fixed value of v along the profile line 504, if the corresponding picket on isocline ribbon 508 is a straight line segment (e.g. line segment 512), the isocline ribbon is a ruled surface as one skilled in the art will understand. Accordingly, the parameter u provides a distance-like measurement along the ruling where the point (u,v) is found. Assume that each of the isocline ribbons 508 and 516 are ruled surfaces. Further assume that the pre-image of each of the profiles 504 and 506 in parameter space are the profiles themselves and the distance-like measurement is the parametric u value of a point (u,v_0) on the v_0 ruling of the isocline. Because the isocline ribbons 508 and 516 are ruled surfaces, for the constant v_0 parameter we can scan out a set of equidistant points along line segments 512 and 520 by simply adding the appropriate offset vector to the previous value. The initial value is $S_i(0,v_0)$. The offset vector is obtained as

$$T_0 = [S_1(1, v_0) - S_1(0, v_0)] / n, \quad (10)$$

where n is the number of points desired on the ruling line to scan from one isocline ribbon (pre-image) edge to the opposing other edge.

If we impose the restriction that the blending functions are a partition of unity, i.e., $B_1 = 1 - B_2$, which is desirable from a design perspective, then the Formula (I) yields

$$S(u, v) = [S_1(u, v) - S_2(u, v)] \cdot B_1(u, v) + S_2(u, v) \quad (II)$$

In one embodiment, this form and with the previous simplifications, it is seen that each point requires three vector adds (for S_1 , S_2 and the “+”), one table look up (for $B_1(u, v)$) and one scalar multiply. This is after initialization which consists of finding each $S_i(0, v)$ and computing T_0 , the offset vector (using Formula (10)). To scan out a set of points on S , one simply increments through the parameter v , and then computes points along the rulings in u .

In the case of the defined four-edge surface (as in Section 4.6.1), some S_i are as the two edge case above, but the others blend longitudinally across the ribbon first. Specifically, in Fig. 34 the v -loft case is the same as Fig. 35 with re-labeling, while the u -loft of Fig. 34 is a horizontal blend of isocline ribbons. The four-edge surface results from the barycentric blend of all four.

In Fig. 33, four profile curves P_{11} , P_{12} , P_{21} and P_{22} are shown between which a surface is desired to be placed. In Fig. 33, the profiles P_{11} and P_{12} together with their corresponding respective isocline ribbons R_{11} and R_{12} , are used to create the blended surface S_1 .

While S_1 evaluates exactly as the two-edge case already described, the generation of S_2 (Fig. 35) differs because the u and v parameters are reversed. In this case, the straight line segment on the isoclines R_{21} and R_{22} corresponds to fixing u and scanning in v ; a situation which is incompatible to rapid scanning. However, it is desirable to fix just one parameter and scan the other in both Figs. 34 and 35. In one embodiment, this can be resolved by defining the isoclines R_{21} and R_{22} differently. That is, each such ribbon becomes a blend of two ruled surfaces defined by simple user inputs. For example, consider isocline ribbon R_{21} . It may be defined by blending two bilinear surfaces 1950 and 1952 in a manner similar to the surface generation techniques described in section 2.2 and illustrated in Figs. 37 through 39. That is, the edges of the bilinear surfaces 1950 and 1952 that are tangent on the profile P_{21} are the profile handles 1956 and 1960; the boundary handles 1964 and 1968 are tangent to the ribbon boundary 1972 and form the opposite edges of the bilinear surfaces. The other two line segments 1976 and 1980 are user inputs.

It is now possible to fix v in the second (u -loft) as well and scan by adding a single vector offset. This effort produces points on the isocline ribbons, S_{21} and S_{22} , each at the identical cost of producing points on the v -loft. Additionally, we must blend the new points to compute the point on S_2 . In operation counts, there are, therefore, eleven vector additions, five scalar multiplies and one table look-up. The additions include three for the v -loft, three each for the u -loft isoclines, one for blending these isoclines and one for blending the two lofts.

For the general N -sided surface it is first necessary to compute a distance on each ribbon. The parameters are calculated using the N -sided parameterization technique from Section 2.2. These distances are then plugged into the blending functions of Formula (6). They are adjusted so they vary from 0 to 1.

The parameters for the ribbons must be set from the distance so given. That is, one parameter will be the distance (from the profile). The other parameter can be deduced by determining where the parameter line of Figure 12 crosses the edge of the N -sided polygon. It is assumed that the polygon has edges of length 1. After these parameters are figured then Formula (4) has all constituents needed for calculation.

6. Applications

The present invention may be used in a large number of computational design domains. In particular, the following list provides brief descriptions of some of the areas where the present invention can be applied.

6.1 Container Design

Free-form design of containers such as bottles has been heretofore non-intuitive and tedious. The present invention alleviates these drawbacks.

6.2 Automotive Design

In the automotive industry, the present invention can be used for auto body design as well as for auto component design. In particular, the ease with which deformations of parts and contours can be performed with the present invention allows

for straightforward deformation of components and recesses so that the fitting of components in particular recesses is more easily designed.

6.3 Aerospace

5 The present invention provides high precision trimming and surface patching operations which are required by the aerospace industry.

6.4 Shipbuilding

Unique to the shipbuilding industry is the need for the design of ship hulls and propellers. Designs of both hulls and propellers may be driven by the physics of the constraints related to water flow. The satisfaction of such constraints can be incorporated into the present invention.

10 6.5 Traditional CAD/CAM Applications

Applications for the design of engines, piping layouts and sheet metal products typically require trimming and blending capabilities. Thus, since the present invention is particularly efficient at providing such operations as well as providing easy deformations of surfaces, its effectiveness in these areas may be of particular merit.

6.6 Other Applications

15 The following is a list of other areas where the present invention may be used for computational design. These are: home electronic and appliance design, plastic injection mold design, tool and die design, toy design, geological modeling, geographical modeling, mining design, art and entertainment, animation, sculpture, fluid dynamics, meteorology, heat flow, electromagnetics, plastic surgery, burn masks, orthodontics, prosthetics, clothing design, shoe design, architectural design, virtual reality design, scientific visualization of data, geometric models for training personnel (e.g., medical training).

20 The foregoing discussion of the invention has been presented for purposes of illustration and description. Further, the description is not intended to limit the invention to the form disclosed herein. Consequently, variations and modifications

commensurate with the above teachings, within the skill and knowledge of the relevant art, are within the scope of the present invention. The embodiment described hereinabove is further intended to explain the best mode presently known of practicing the invention and to enable others skilled in the art to utilize the invention as such, or in other embodiments, and with the various modifications required by their particular application or uses of the invention. It is intended that the appended claims be

5 construed to include alternative embodiments to the extent permitted by the prior art.